



TUGAS AKHIR - TE 141599

**DESAIN KONTROLER PID-*GENETIC ALGORITHM*
UNTUK SISTEM PENGATURAN LEVEL AIR *STEAM*
DRUM PADA PEMBANGKIT LISTRIK TENAGA UAP
(PLTU)**

Mohamad Yusuf
NRP. 2213100086

Dosen Pembimbing
Ir. Ali Fatoni, MT.
Mohamad Abdul Hady, ST., MT.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2017



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE 141599

***PID-GENETIC ALGORITHM CONTROLLER DESIGN FOR
WATER LEVEL CONTROL SYSTEM OF STEAM DRUM
IN STEAM POWER PLANT***

Mohamad Yusuf
NRP. 2213100086

Supervisor
Ir. Ali Fatoni, MT.
Mohamad Abdul Hady, ST., MT.

***DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017***

PERNYATAAN KEASLIAN TUGAS AKHIR

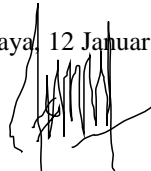
Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul :

**“Desain Kontroler PID-*Genetic Algorithm*
untuk Sistem Pengaturan Level Air *Steam Drum*
pada Pembangkit Listrik Tenaga Uap (PLTU)”**

adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 12 Januari 2017



Mohamad Yusuf
NRP. 2213100086

Halaman ini sengaja dikosongkan

**DESAIN KONTROLER PID-GENETIC ALGORITHM
UNTUK SISTEM PENGATURAN LEVEL AIR STEAM
DRUM PADA PEMBANGKIT LISTRIK TENAGA UAP
(PLTU)**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan Untuk
Memperoleh Gelar Sarjana Teknik Elektro
Pada**


**Bidang Studi Teknik Sistem Pengaturan
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui

Dosen Pembimbing I,


Ir. Ali Fatoni, MT.
NIP. 196206031989031002

Dosen Pembimbing II,


Mohamad Abdul Hady, ST., MT.
NIP. 198904132015041002



Halaman ini sengaja dikosongkan

**DESAIN KONTROLER PID-*GENETIC ALGORITHM*
UNTUK SISTEM PENGATURAN LEVEL AIR *STEAM DRUM*
PADA PEMBANGKIT LISTRIK TENAGA UAP (PLTU)**

Mohamad Yusuf
2213100086

Dosen Pembimbing I	: Ir. Ali Fatoni, MT.
NIP	: 196206031989031002
Dosen Pembimbing II	: Mohamad Abdul Hady, ST., MT.
NIP	: 198904132015041002

ABSTRAK

Perubahan laju aliran uap menimbulkan gangguan pada sistem pengaturan level air *steam drum*. Level air dijaga pada titik tengah ketinggian drum atau disebut *Normally Water Level* (NWL) agar uap yang dihasilkan sesuai spesifikasi serta tidak merusak peralatan. Pada umumnya, sistem pengaturan level menggunakan kontroler PID konvensional. Namun, adanya gangguan menyebabkan performa sistem dengan kontroler PID konvensional tidak mampu memenuhi spesifikasi. Kontroler PID-*Genetic Algorithm* (PID-GA) diterapkan untuk mengatur level air *steam drum* agar berada pada titik NWL ketika terdapat gangguan. Sistem pengaturan menggunakan kontroler PID-GA mampu meredam gangguan berupa beban minimal, nominal, dan maksimal, yaitu dengan *perturbation peak* masing-masing 0,18 m; 0,22 m; dan 0,26 m.

Kata Kunci : *Genetic Algorithm*, PID, PLTU, *Steam Drum*.

Halaman ini sengaja dikosongkan

***PID-GENETIC ALGORITHM CONTROLLER DESIGN
FOR STEAM DRUM WATER LEVEL CONTROL SYSTEM
IN STEAM POWER PLANT***

Mohamad Yusuf
2213100086

Supervisor I : Ir. Ali Fatoni, MT.
ID Number : 196206031989031002
Supervisor II : Mohamad Abdul Hady, ST., MT.
ID Number : 198904132015041002

ABSTRACT

Changes in steam flow rate causing harm to steam drum water level control system. The water level is maintained at the midpoint of the height of the drum or referred Normally Water Level (NWL) so that the steam produced according to specifications and does not damage the equipments. In general, level control using the conventional PID controller. However, the disturbances caused by the system performance of conventional PID controller is not able to meet the specifications. PID- Genetic Algorithm controller (PID-GA) was applied to adjust the steam drum water level in order to be at a point when there is a perturbation NWL. System settings using the PID controller-GA is able to reduce disturbance of the minimum, nominal, and maximum with perturbation peak each of them are 0.18 m, 0.22 m, and 0.26 m.

Keywords: Genetic Algorithm, PID, Steam Power Plant, Steam Drum.

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan syukur kami panjatkan kehadiran Allah SWT, karena berkat rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan penelitian dengan judul “**Desain Kontroler PID-Genetic Algorithm untuk Sistem Pengaturan Level Air Steam Drum pada Pembangkit Listrik Tenaga Uap (PLTU)**” untuk memenuhi syarat kelulusan pada Bidang Studi Teknik Sistem Pengaturan, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Laporan penelitian ini dapat diselesaikan oleh penulis berkat bantuan, bimbingan, dan dukungan dari berbagai pihak. Penulis ingin mengucapkan terima kasih kepada Ibu, Bapak, dan Kakak tercinta yang selalu memberi dukungan, semangat, dan doa untuk keberhasilan penulis. Bapak Ir. Ali Fatoni, MT. dan Bapak Mohamad Abdul Hady, ST., MT. selaku dosen pembimbing atas bimbingan dan arahannya. Bapak Imam Arifin, ST., MT., dan Bapak Ir. Joko Susila, MT. selaku pembimbing laboratorium yang telah mendidik penulis menjadi lebih baik. Dosen Bidang Studi Teknik Sistem Pengaturan dan Jurusan Teknik Elektro atas pendidikan dan ilmunya. Bapak karyawan PT. Pembangkitan Jawa-Bali (PJB) Unit Pembangkitan (UP) 1 dan 2 Paiton divisi Har. Kontrol dan Instrumen atas ilmu dan pengalamannya. Keluarga laboratorium AJ-104 atas bantuan dan kerja samanya. Teman-teman dan semua pihak yang tidak dapat penulis sebutkan satu persatu.

Penulis berharap laporan ini dapat bermanfaat bagi pembaca pada umumnya dan penulis pada khususnya. Laporan ini masih jauh dari sempurna, sehingga penulis mengharapkan kritik dan saran dari pembaca yang bersifat membangun.

Surabaya, 07 Januari 2017

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN KEASLIAN TUGAS AKHIR	iii
LEMBAR PENGESAHAN.....	v
ABSTRAK.....	vii
ABSTRACT.....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL	xx
BAB 1 PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian	3
1.5 Metodologi.....	3
1.6 Sistematika Penulisan.....	3
1.7 Relevansi.....	4
BAB 2 SISTEM PENGATURAN LEVEL AIR STEAM DRUM PADA PLTU	
2.1 Pembangkit Listrik Tenaga Uap (PLTU)	5
2.2 Sistem Pengaturan Level Air <i>Steam Drum</i>	6
2.2.1 <i>One-element Level Control</i> pada <i>Steam Drum</i>	6
2.2.2 <i>Pemodelan Steam Drum</i>	7
2.2.3 <i>Pemodelan Fluid Coupling</i>	10
2.2.4 <i>Pemodelan Pompa Boiler</i>	10
2.2.5 <i>Pemodelan Pipa</i>	11
2.2.6 <i>Pemodelan Level Transmitter</i>	11
2.2.7 <i>Pendekatan Padè</i>	11
2.3 Kontroler PID.....	12
2.3.1 Aksi Proporsional	13
2.3.2 Aksi Integral	13
2.3.3 Aksi Derivatif	14
2.4 Aturan <i>Tuning</i> Kontroler PID Ziegler-Nichols.....	14
2.4.1 Metode <i>Open Loop</i>	15
2.4.2 Metode <i>Closed Loop</i>	16

2.5	Indeks Performansi Kesalahan.....	17
2.5.1	<i>Integral Square Error (ISE)</i>	17
2.5.2	<i>Integral of Absolute Value of Error (IAE)</i>	17
2.5.3	<i>Integral of Time Multiplied by Absolute Value of Error (ITAE)</i>	18
2.6	<i>Genetic Algorithm (GA)</i>	18
2.6.1	Representasi dan Inisialisasi Populasi	19
2.6.2	Fungsi Objektif dan Fungsi <i>Fitness</i>	20
2.6.3	Seleksi.....	20
2.6.4	<i>Crossover</i>	21
2.6.5	Mutasi	22
2.7	Kontroler PID- <i>Genetic Algorithm</i> (PID-GA)	22
BAB 3	SIMULATOR PLTU DAN FEEDWATER SYSTEM	
3.1	Perancangan Simulator PLTU	25
3.2	Arsitektur Simulator PLTU	26
3.2.1	Jaringan Komunikasi	27
3.2.2	HMI <i>Feedwater Systems Overview</i>	27
3.2.3	HMI <i>Boiler Feed Pumps (Systems)</i>	28
3.2.4	HMI <i>Boiler Feed Pumps (Components)</i>	29
3.2.5	HMI BFP <i>Sequences</i>	29
3.2.6	HMI BFP <i>Permissives</i>	30
3.2.7	HMI <i>High Pressure Heaters (HPH)</i>	30
3.3	Pemodelan Sistem Pengaturan Level Air <i>Steam Drum</i>	31
3.4	Spesifikasi Respon	33
3.5	Desain Kontroler PID Ziegler-Nichols (PID-ZN)	34
3.6	Desain Kontroler PID-GA	35
BAB 4	PENGUJIAN DAN ANALISIS METODE KONTROL PID DAN PID-GA	
4.1	Deskripsi Beban pada Pengujian Metode Kontrol	37
4.1.1	Beban Tetap.....	37
4.1.2	Beban Acak	38
4.1.3	Beban dengan Perubahan Kriteria	39
4.2	Pengujian Sistem secara <i>Open Loop</i>	39
4.3	Pengujian Sistem secara <i>Closed Loop</i>	41
4.4	Pengujian Sistem dengan Kontroler PID.....	44
4.4.1	Pengujian dengan Variasi Nilai K_p	45
4.4.2	Pengujian dengan Variasi Nilai τ_i	46
4.4.3	Pengujian dengan Variasi Nilai τ_d	47
4.4.4	Pengujian dengan Beban Tetap	48

4.4.5	Pengujian dengan Beban Acak	50
4.4.6	Pengujian dengan Perubahan Kriteria Beban	51
4.5	Pengujian Sistem dengan Kontroler PID-GA	52
4.5.1	Pengujian dengan Beban Tetap	53
4.5.2	Pengujian dengan Beban Acak	55
4.5.3	Pengujian dengan Perubahan Kriteria Beban	57
4.5.4	Pengujian dengan Variasi Fungsi Objektif	59
4.5.5	Pengujian dengan Variasi Jumlah Generasi	60
4.5.6	Pengujian dengan Variasi Probabilitas <i>Crossover</i>	62
4.5.7	Pengujian dengan Variasi Probabilitas Mutasi	64
BAB 5 KESIMPULAN		67
DAFTAR PUSTAKA		69
LAMPIRAN		71
	Lampiran 1. <i>Update</i> Parameter dengan Variasi Jumlah Generasi	71
	Lampiran 2. <i>Update</i> Parameter dengan Variasi Prob. <i>Crossover</i>	72
	Lampiran 3. <i>Update</i> Parameter dengan Variasi Probabilitas Mutasi	73
	Lampiran 4. Program Kontroler PID-GA	74
	Lampiran 5. Program Indeks Performansi Kesalahan	77
	Lampiran 6. Simulink Kontroler PID-GA	78
RIWAYAT PENULIS		83

Halaman ini sengaja dikosongkan

TABLE OF CONTENT

COVER PAGE	i
AUTHENTICITY	iii
APPROVAL SHEET	v
ABSTRAK	vii
ABSTRACT	ix
PREFACE	xi
TABLE OF CONTENT	xiii
LIST OF FIGURES	xvii
LIST OF TABLES	xx
CHAPTER 1 INTRODUCTION	
1.1 Background	1
1.2 Problems	2
1.3 Problem Constraints.....	3
1.4 Research Objectives	3
1.5 Methodology	3
1.6 Systematics	3
1.7 Relevances	4
CHAPTER 2 STEAM DRUM WATER LEVEL CONTROL	
SYSTEMS IN STEAM POWER PLANT	
2.1 Steam Power Plant	5
2.2 Steam Drum Water Level Control System	6
2.2.1 One-element Level Control in Steam Drum	6
2.2.2 Steam Drum Modelling.....	7
2.2.3 Fluid Coupling Modelling.....	10
2.2.4 Boiler Pump Modelling.....	10
2.2.5 Pipe Modelling	11
2.2.6 Level Transmitter Modelling	11
2.2.7 Padé Approximations.....	11
2.3 PID Controller.....	12
2.3.1 Proportional Action	13
2.3.2 Integral Action.....	13
2.3.3 Derivative Action	14
2.4 PID Controller Tuning using Ziegler-Nichols Rule.....	14
2.4.1 Open Loop Method.....	15
2.4.2 Closed Loop Method	16
2.5 Error Performance Indices	17
2.5.1 Integral Square Error (ISE).....	17

2.5.2	Integral of Absolute Value of Error (IAE).....	17
2.5.3	Integral of Time Multiplied by Absolute Value of Error (ITAE).....	18
2.6	Genetic Algorithm (GA).....	18
2.6.1	Representation and Initialization of Population	19
2.6.2	Objective Function and Fitness Function	20
2.6.3	Selection.....	20
2.6.4	Crossover	21
2.6.5	Mutation	22
2.7	PID-Genetic Algorithm Controller (PID-GA).....	22
BAB 3	STEAM POWER PLANT SIMULATOR AND	
	FEEDWATER SYSTEM	
3.1	Steam Power Plant Simulator Design	25
3.2	Steam Power Plant Simulator Architecture	26
3.2.1	Communication Network.....	27
3.2.2	Feedwater Systems Overview HMI.....	27
3.2.3	Boiler Feed Pumps (Systems) HMI.....	28
3.2.4	Boiler Feed Pumps (Components) HMI	29
3.2.5	BFP Sequences HMI.....	29
3.2.6	BFP Permissives HMI	30
3.2.7	High Pressure Heaters (HPH) HMI	30
3.3	Steam Drum Water Level Control System Modelling	31
3.4	Response Specifications	33
3.5	PID Ziegler-Nichols Controller Design(PID-ZN)	34
3.6	PID-GA Controller Design	35
BAB 4	SIMULATIONS AND PID AND PID-GA CONTROLLER	
	ANALYSIS	
4.1	Load Criterion in Controller Simulations.....	37
4.1.1	Fixed Load	37
4.1.2	Random Load	38
4.1.3	Load with Changes of Criterion	39
4.2	Open Loop Simulations	39
4.3	Closed Loop Simulations.....	41
4.4	Simulations with PID Controller.....	44
4.4.1	Simulation with Variation of K_p	45
4.4.2	Simulation with Variation of τ_i	46
4.4.3	Simulation with Variation of τ_d	47
4.4.4	Simulation with Fixed Load.....	48
4.4.5	Simulation with Random Load.....	50

4.4.6	Simulation with Changes of Load Criterion	51
4.5	Simulation with PID-GA Controller	52
4.5.1	Simulation with Fixed Load	53
4.5.2	Simulation with Random Load	55
4.5.3	Simulation with Changes of Load Criterion	57
4.5.4	Simulation with Variation of Objective Function	58
4.5.5	Simulation with Variation of Generation Number	60
4.5.6	Simulation with Variation of Crossover Probability	62
4.5.7	Simulation with Variation of Mutation Probability	64
CHAPTER 5 CONCLUSIONS		67
BIBLIOGRAPGY		69
APPENDIXES		71
Appendix 1. Parameter Update with Variation of Generation Num .		71
Appendix 2. Parameter Update with Variation of Crossover Prob ...		72
Appendix 3. Parameter Update with Variationof Mutation Prob		73
Appendix 4. PID-GA Controller Program		74
Appendix 5. Error Performance Indices Program		77
Appendix 6. PID-GA Controller Simulink		78
BIOGRAPHY		83

DAFTAR GAMBAR

Gambar 2.1. Perubahan Energi pada PLTU	5
Gambar 2.2. Diagram Blok Sistem Pengaturan Level Air <i>Steam Drum</i> ...	6
Gambar 2.3. P&ID Sistem Pengaturan Level Air <i>Steam Drum</i>	7
Gambar 2.4. Tampilan Fisik dari <i>Steam Drum</i>	8
Gambar 2.5. Representasi Skema Pemodelan <i>Steam Drum</i>	8
Gambar 2.6. Respon Sistem dengan Bentuk Huruf-S.....	15
Gambar 2.7. Respon Osilasi Permanen dengan Periode P_{cr}	16
Gambar 2.8. Representasi Kromosom pada GA.....	19
Gambar 2.9. Diagram Alir GA	19
Gambar 2.10. Representasi Kromosom dengan Bilangan Biner	20
Gambar 2.11. Mekanisme Seleksi Tipe <i>Tournament</i>	21
Gambar 2.12. Mekanisme <i>Single-point Crossover</i>	21
Gambar 2.13. Mekanisme <i>Flip Mutation</i>	22
Gambar 2.14. Diagram Blok Sistem dengan Kontroler PID-GA	23
Gambar 2.15. Diagram Alir <i>Online Tuning</i> Kontroler PID-GA	23
Gambar 3.1. Arsitektur Simulator PLTU	26
Gambar 3.2. Tampilan HMI <i>Feedwater Systems Overview</i>	28
Gambar 3.3. Tampilan HMI <i>Boiler Feed Pumps (Systems)</i>	28
Gambar 3.4. Tampilan HMI <i>Boiler Feed Pumps (Components)</i>	29
Gambar 3.5. Tampilan HMI <i>BFP Sequences</i>	29
Gambar 3.6. Tampilan HMI <i>BFP Permissives</i>	30
Gambar 3.7. Tampilan HMI <i>High Pressure Heater (HPH)</i>	30
Gambar 3.8. Representasi Skema <i>Steam Drum</i>	31
Gambar 3.9. Diagram Blok Sistem Pengaturan <i>Closed Loop</i>	32
Gambar 3.10. Spesifikasi Respon Level Air <i>Steam Drum</i>	33
Gambar 3.11. Diagram Blok <i>Tuning</i> PID-ZN Metode <i>Closed Loop</i>	34
Gambar 3.12. Respon Sistem dengan $K_p = 8,78$	35
Gambar 3.13. Representasi Individu GA	35
Gambar 4.1. Representasi Perubahan Beban Tetap	38
Gambar 4.2. Representasi Perubahan Beban Acak.....	38
Gambar 4.3. Representasi Perubahan Beban Berubah.....	39
Gambar 4.4. Diagram Blok Pengujian Sistem <i>Open Loop</i>	40
Gambar 4.5. Respon Level dengan Masukan Sinyal Uji <i>Step</i>	40
Gambar 4.6. Diagram Blok Pengujian Sistem <i>Closed Loop</i>	41
Gambar 4.7. Respon Sistem <i>Closed Loop</i> dengan $K_p = 1$, $R_s = 0$, dan $D_s = 2,12$	42
Gambar 4.8. Letak <i>Pole</i> dan <i>Zero</i> Sistem	43

Gambar 4.9. Diagram Blok Sistem Pengaturan dengan Kontroler PID..	44
Gambar 4.10. Respon Sistem Menggunakan Kontroler PID dengan Variasi K_p	45
Gambar 4.11. Respon Sistem Menggunakan Kontroler PID dengan Variasi τ_i	47
Gambar 4.12. Respon Sistem Menggunakan Kontroler PID dengan Variasi τ_d	48
Gambar 4.13. Respon Sistem Menggunakan Kontroler PID dengan Beban Tetap	49
Gambar 4.14. Respon Sistem Menggunakan Kontroler PID dengan Beban Acak	51
Gambar 4.15. Respon Sistem Menggunakan Kontroler PID dengan Beban Berubah	52
Gambar 4.16. Diagram Blok Sistem Pengaturan Menggunakan Kontrol PID-GA	52
Gambar 4.17. Respon Menggunakan Kontroler PID-GA dengan Beban Tetap	54
Gambar 4.18. Nilai <i>Fitness</i> pada Kontroler PID-GA dengan Beban Tetap	55
Gambar 4.19. Respon Menggunakan Kontroler PID-GA dengan Beban Acak	56
Gambar 4.20. Respon Menggunakan Kontroler PID dan PID-GA dengan Beban Acak	56
Gambar 4.21. Respon Menggunakan Kontroler PID-GA dengan Beban Berubah	58
Gambar 4.22. Respon Menggunakan Kontroler PID dan PID-GA dengan Beban Berubah	58
Gambar 4.23. Respon Level Air <i>Steam Drum</i> Menggunakan Kontroler PID-GA dengan Variasi Fungsi Objektif	59
Gambar 4.24. Nilai <i>Fitness</i> pada Kontroler PID-GA dengan Variasi Fungsi Objektif	60
Gambar 4.25. Respon Level Air <i>Steam Drum</i> Menggunakan Kontroler PID-GA dengan Variasi Jumlah Generasi	61
Gambar 4.26. <i>Update</i> Parameter Kontroler PID-GA dengan Variasi Jumlah Generasi	62
Gambar 4.27. Respon Level Air <i>Steam Drum</i> Menggunakan Kontroler PID-GA dengan Variasi Probabilitas <i>Crossover</i>	63
Gambar 4.28. Nilai <i>Fitness</i> dengan Variasi Probabilitas <i>Crossover</i>	64

Gambar 4.29. <i>Update</i> Parameter Kontroler PID-GA dengan Variasi Probabilitas <i>Crossover</i>	64
Gambar 4.30. Respon Level Air <i>Steam Drum</i> Menggunakan Kontroler PID-GA dengan Variasi Probabilitas Mutasi	65
Gambar 4.31. <i>Update</i> Parameter Kontroler PID-GA dengan Variasi Probabilitas Mutasi.....	66

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 2.1. Variabel pada Model Matematis <i>Steam Drum</i>	9
Tabel 2.2. Aturan <i>Tuning</i> Kontroler PID Ziegler-Nichols Metode <i>Open Loop</i>	15
Tabel 2.3. Aturan <i>Tuning</i> Kontroler PID Ziegler-Nichols Metode <i>Closed Loop</i>	16
Tabel 3.1. Parameter pada Model Matematis <i>Steam Drum</i>	32
Tabel 3.2. Nilai Parameter Kontroler PID-ZN Metode <i>Closed Loop</i>	34
Tabel 4.1. Kriteria Beban Pembangkit dan Laju Aliran Uap	37
Tabel 4.2. Spesifikasi Respon Menggunakan Kontroler PID dengan Variasi K_p	46
Tabel 4.3. Spesifikasi Respon Menggunakan Kontroler PID dengan Variasi τ_i	47
Tabel 4.4. Spesifikasi Respon Menggunakan Kontroler PID dengan Variasi τ_d	48
Tabel 4.5. Spesifikasi Respon Menggunakan Kontroler PID dengan Beban Tetap	50
Tabel 4.6. Spesifikasi Respon Menggunakan Kontroler PID dengan Beban Acak	50
Tabel 4.7. Spesifikasi Respon Menggunakan Kontroler PID dengan Beban Berubah.....	51
Tabel 4.8. Parameter yang Digunakan pada GA	53
Tabel 4.9. Spesifikasi Respon Sistem Menggunakan Kontroler PID-GA dengan Beban Tetap	55
Tabel 4.10. Spesifikasi Respon Sistem Menggunakan Kontroler PID-GA dan PID dengan Beban Acak	57
Tabel 4.11. Spesifikasi Respon Sistem Menggunakan Kontroler PID-GA dan PID dengan Beban Berubah	58
Tabel 4.12. Spesifikasi Respon Sistem Menggunakan Kontroler PID-GA dengan Variasi Fungsi Objektif	60
Tabel 4.13. Spesifikasi Respon Sistem Menggunakan Kontroler PID-GA dengan Variasi Jumlah Generasi	62
Tabel 4.14. Spesifikasi Respon Sistem Menggunakan Kontroler PID-GA dengan Variasi Probabilitas <i>Crossover</i>	63
Tabel 4.15. Spesifikasi Respon Sistem Menggunakan Kontroler PID-GA dengan Variasi Probabilitas Mutasi	65

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

Latar belakang dilakukan penelitian tentang kontroler PID-GA pada sistem pengaturan level air *steam drum*, berawal dari upaya meredam gangguan pada sistem tersebut. Gangguan pada sistem pengaturan level air *steam drum*, dapat mengakibatkan kerusakan komponen pembangkit. Permasalahan pada penelitian ini, dibatasi oleh model matematis, tipe pengaturan, jenis GA, dan fungsi objektif yang digunakan. Kontroler PID-GA digunakan untuk memperbaiki respon sistem ketika terjadi gangguan. Penelitian diawali dengan melakukan studi literatur, memodelkan sistem, mendesain kontroler PID dan PID-GA. Sistematika penulisan laporan penelitian ini terdiri dari 5 Bab.

1.1 Latar Belakang

PLTU merupakan jenis instalasi pembangkit listrik yang menggunakan tenaga potensial uap untuk memutar turbin sebagai penggerak utama (*prime mover*). Uap berasal dari air yang dipanaskan pada *boiler*. *Steam drum* merupakan bagian dari *boiler* yang berfungsi untuk menampung dan memisah campuran antara air dan uap. Ketika permintaan daya listrik meningkat, turbin membutuhkan tambahan uap untuk menjaga batas perubahan frekuensi sebesar $\pm 0,5$ Hz [1]. Laju aliran uap yang keluar dari *steam drum* dapat menimbulkan gangguan berupa naik atau turunnya level air dari kondisi normal atau sering disebut sebagai titik *Normally Water Level* (NWL) [2].

Level air *steam drum* merupakan variabel yang mempengaruhi kondisi operasi *boiler*. Level air yang berlebih akan mempengaruhi proses pemisahan uap dan air, meningkatkan kelembapan uap, dan menimbulkan kerak, serta mengurangi kualitas uap. Level air yang terlalu rendah dapat merusak siklus uap dan memicu terjadinya keretakan pipa *boiler*. Level air harus diatur sesuai dengan ketentuan pembuat *steam drum* agar mampu menghasilkan uap sesuai spesifikasi [3].

Pada umumnya, sistem pengaturan di industri menggunakan kontroler PID, karena memiliki struktur sederhana, mudah diaplikasikan, dan mudah digunakan [4]. Contohnya, sistem pengaturan level air *steam drum* pada PT. Pembangkitan Jawa-Bali (PJB) Unit Pembangkitan (UP) 1 dan 2 Paiton menggunakan kontroler PID [5].

Penelitian tentang metode pengaturan untuk meningkatkan performa sistem telah dilakukan oleh beberapa peneliti. Pada tahun 2008, Ayla Altınten dkk melakukan penelitian tentang penentuan parameter kontroler PID menggunakan *genetic algorithm* (GA). Skema pengaturan *self-tuning* digunakan pada penelitian tersebut. Berdasarkan hasil implementasinya untuk kontrol temperatur pada reaktor *polystyrene*, metode tersebut mampu memenuhi spesifikasi yang diinginkan [6].

Metode pengaturan PID berbasis GA diusulkan untuk melakukan optimisasi *tuning* parameter kontroler PID pada proses *continuous stirred tank reactor* (CSTR). Berdasarkan hasil simulasi yang dilakukan oleh A. Jayachitra dan R. Vinodha pada tahun 2014, diperoleh kesimpulan bahwa *tuning* kontroler PID berbasis GA mampu memenuhi spesifikasi pada kasus *setpoint tracking* dan *disturbance rejection* [7].

Penerapan GA pada *autotuning* parameter kontroler PID telah dilakukan oleh T.K. Teng, J.S. Chieh dan C.S. Chen pada sistem pengaturan level cairan. Metode yang diusulkan mampu menentukan nilai parameter PID yang optimal secara otomatis setiap generasi, sehingga level cairan dapat memenuhi *setpoint*, ketika parameter *plant* berubah [8].

Besarnya kebutuhan laju aliran uap untuk memutar turbin pada pembangkit berbeda setiap waktu, sehingga gangguan pada sistem tersebut berubah-ubah. Sistem pengaturan level air *steam drum* diharapkan mampu mengatasi efek gangguan selama pembangkit beroperasi. Oleh karena itu, pada penelitian ini dilakukan desain kontroler PID-GA untuk mengatur level air *steam drum* agar mampu menjaga level air pada nilai *setpoint* ketika terdapat gangguan. Kontroler PID-GA yang dirancang mampu melakukan *tuning* parameter kontroler secara *online*.

1.2 Perumusan Masalah

Konsumen menggunakan energi listrik berbeda setiap waktu, sehingga menimbulkan fluktuasi beban pada pembangkit. Jika beban pembangkit meningkat, maka laju aliran uap untuk memutar turbin pada pembangkit semakin besar. Hal ini menyebabkan air pada *steam drum* berkurang, sehingga level airnya turun. Jika kontroler tidak dapat menjaga level air pada batas yang diizinkan, maka sistem proteksi pembangkit akan bekerja untuk menghentikan operasi *boiler*. Berdasarkan hal tersebut, pembangkit membutuhkan sistem pengaturan untuk mengatur level air *steam drum* pada nilai *setpoint*.

Level air *steam drum* berubah-ubah berdasarkan laju aliran uap yang keluar dari drum. Oleh karena itu, sistem pengaturan level air *steam drum*

membutuhkan kontroler dengan kemampuan meredam gangguan yang baik agar dapat memenuhi spesifikasi yang diinginkan.

1.3 Batasan Masalah

Penelitian ini menggunakan model matematis dari *first principle* untuk menguji metode pengaturan yang diusulkan. Parameter model yang digunakan diperoleh dari data pembangkit di PT. PJB UP 1 dan 2 Paiton. Tipe pengaturan level air *steam drum* yang digunakan dalam penelitian ini adalah *one-element level control*. Fungsi objektif yang digunakan pada GA adalah kombinasi *Integral Square Error* (ISE), *Integral of Absolute Value of Error* (IAE), dan *Integral of Time Multiplied by Absolute Value of Error* (ITAE).

1.4 Tujuan Penelitian

Desain kontroler PID-GA diterapkan pada sistem pengaturan level air *steam drum* untuk menjaga level air berada pada nilai *setpoint*, ketika terdapat gangguan berupa laju aliran uap keluar dari drum.

1.5 Metodologi

Penelitian ini dilakukan melalui beberapa tahap, diantaranya studi literatur, pemodelan *steam drum*, perancangan kontroler PID, perancangan mekanisme *update* parameter kontroler PID-GA, pengujian dan analisis serta penulisan buku laporan penelitian. Studi literatur dilakukan untuk mengumpulkan teori penunjang dari buku, jurnal, dan data teknik perusahaan. Fungsi alih *plant* dan spesifikasi yang diinginkan menjadi dasar perancangan kontroler PID dan PID-GA. Pengujian dilakukan dengan menganalisis performa sistem tanpa kontroler, konfigurasi *closed loop*, dengan kontroler PID, dan dengan kontroler PID-GA. Kontroler diterapkan pada simulator sebagai media pembelajaran tentang sistem pengaturan level air *steam drum* serta aplikasi kecerdasan buatan pada sistem pengaturan di industri.

1.6 Sistematika Penulisan

Penulisan laporan penelitian ini dapat dijelaskan dengan sistematika sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini membahas tentang latar belakang, permasalahan, batasan masalah, tujuan penelitian, metodologi, sistematika penulisan dan relevansi dari penelitian yang dilakukan.

BAB 2 SISTEM PENGATURAN LEVEL AIR STEAM DRUM PADA PLTU

Konsep dan teori yang mendasari penelitian ini meliputi PLTU, sistem pengaturan level air *steam drum*, kontroler PID, aturan *tuning* Ziegler-Nichols, indeks performansi kesalahan, *genetic algorithm*, dan kontroler PID-GA dijelaskan pada bab ini.

BAB 3 SIMULATOR PLTU DAN FEEDWATER SYSTEMS

Perancangan simulator dan sistem seperti *virtual plant*, *Human Machine Interface* (HMI), pemodelan sistem, dan perancangan kontroler yang diimplementasikan pada simulator PLTU dijelaskan pada Bab 3.

BAB 4 PENGUJIAN DAN ANALISIS

Bab ini membahas tentang hasil simulasi dari kontroler PID-*Genetic Algorithm* pada simulator PLTU dan analisis performa sistem terhadap gangguan.

BAB 5 KESIMPULAN

Bab 5 membahas tentang kesimpulan berkaitan dengan penelitian yang dilakukan.

1.7 Relevansi

Gangguan pada sistem pengaturan level cairan sering ditemukan di industri. Kontroler diharapkan mampu mengatasi terjadinya gangguan agar respon sistem memenuhi spesifikasi yang diinginkan. Pada penelitian ini didesain kontroler PID yang mampu mengatasi gangguan melalui mekanisme *tuning* parameter menggunakan GA. Hasil desain dapat menjadi media pembelajaran untuk menggambarkan sistem pengaturan level cairan pada *real plant* di industri. Selain itu, dapat digunakan sebagai media pembelajaran aplikasi kecerdasan buatan pada sistem pengaturan di industri.

BAB 2

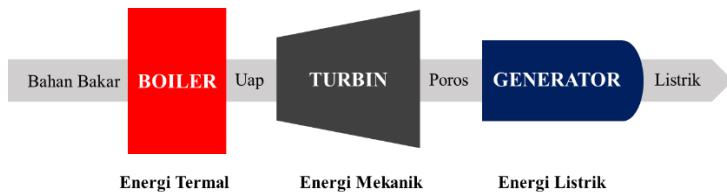
SISTEM PENGATURAN LEVEL AIR *STEAM DRUM* PADA PLTU

PLTU merupakan jenis pembangkit yang menggunakan bahan bakar utama berupa batu bara. Sistem pengaturan level air *steam drum* memiliki komponen seperti *fluid coupling*, pompa *boiler*, pipa, dan *level transmitter*. Kontroler PID banyak digunakan untuk melakukan pengaturan proses di industri. Nilai parameter kontroler PID dapat di-*tuning* menggunakan aturan Ziegler-Nichols. Indeks performansi kesalahan dapat digunakan untuk menyatakan spesifikasi respon sistem.

GA merupakan metode pencarian acak yang meniru mekanisme evolusi makhluk hidup, yaitu melibatkan proses seleksi alam dan operasi genetika. Kontroler PID-GA merupakan pengembangan kontroler PID dengan menambahkan mekanisme *tuning* dengan GA untuk memperoleh parameter kontroler yang optimal terhadap fungsi objektif.

2.1 Pembangkit Listrik Tenaga Uap (PLTU) [9]

PLTU merupakan jenis instalasi pembangkit yang memanfaatkan energi kinetik uap untuk memproduksi energi listrik. Proses perubahan energi pada pembangkit jenis ini berlangsung melalui tiga tahapan, yaitu energi termal menjadi energi mekanik, kemudian energi mekanik menjadi energi listrik. Proses perubahan tersebut diilustrasikan pada Gambar 2.1.



Gambar 2.1. Perubahan Energi pada PLTU

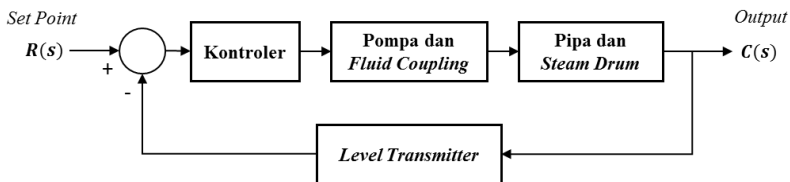
Prinsip kerja PLTU menggunakan sistem pembakaran batu bara untuk memanaskan air sebagai penghasil uap yang akan memutar turbin dan generator. Uap yang telah digunakan untuk memutar turbin diubah wujudnya menjadi air melalui proses kondensasi untuk dipanaskan kembali. Siklus tersebut berlangsung terus-menerus sehingga membentuk

siklus tertutup. Sistem pembangkit tersebut dalam memproduksi listrik dapat dikelompokkan menjadi empat proses utama sebagai berikut:

- 1) Sistem penanganan batu bara dan Abu (*Coal and Ash Handling System*)
- 2) Sistem pengolahan air (*Water Treatment System*)
- 3) Sistem air dan uap (*Water and Steam System*)
- 4) Sistem udara dan gas (*Air and Gas System*)

2.2 Sistem Pengaturan Level Air *Steam Drum* [5]

Energi listrik yang diproduksi oleh pembangkit bergantung pada jumlah uap yang dihasilkan *boiler*. Uap diperoleh dari proses pemisahan campuran air dan uap pada *steam drum* yang terjadi pada sistem air dan uap. Level air *steam drum*, dipengaruhi oleh laju aliran uap yang digunakan untuk memutar turbin. Diagram blok sistem pengaturan level air *steam drum* dapat dilihat pada Gambar 2.2.



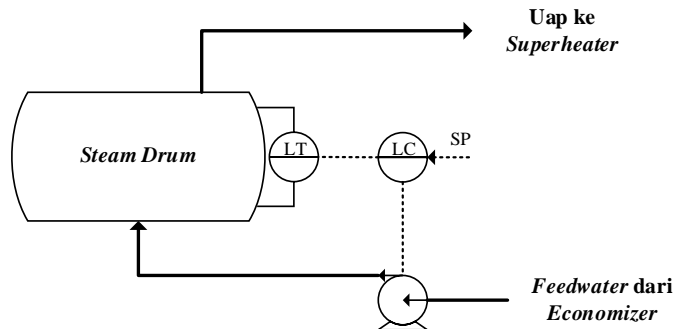
Gambar 2.2. Diagram Blok Sistem Pengaturan Level Air *Steam Drum*

Level air *steam drum* dijaga nilainya agar berada pada nilai sesuai data perusahaan pembuat komponen *boiler* tersebut. Jika level air melebihi batas normalnya, maka air akan terbawa oleh uap menuju *superheater* yang dapat menyebabkan kerusakan pada turbin. Kerusakan pada *boiler* atau turbin membutuhkan biaya pemeliharaan yang mahal, sehingga dapat merugikan perusahaan. Jika level air terlalu rendah, maka dapat menyebabkan pipa-pipa *boiler* pecah serta kegagalan pembangkit gagal beroperasi. Ketika level air *steam drum* terlalu rendah dan melebihi batas, maka *boiler* akan berhenti bekerja (*trip*) untuk mencegah kerusakan pipa yang terhubung dengan *steam drum*.

2.2.1 One-element Level Control pada *Steam Drum* [10]

Sistem pengaturan level air *steam drum* dengan *one-element* dapat dijelaskan melalui *Piping & Instrumentation Diagram* (P&ID) pada

Gambar 2.3. Nilai level air aktual pada drum diukur oleh *level transmitter* dan dikirimkan berupa sinyal tegangan menuju *process control unit* (PCU) untuk dibandingkan dengan nilai *setpoint*. Jika terdapat perbedaan nilai antara level aktual dan *setpoint*, maka terdapat kesalahan pada sistem tersebut. Nilai kesalahan diproses oleh kontroler level untuk menentukan aksi kontrol yang harus dilakukan untuk menjaga level air pada nilai *setpoint* yang ditentukan. Sinyal kontrol dikirimkan menuju aktuator dalam kasus ini adalah *fluid coupling* dan pompa *boiler* untuk menambah atau mengurangi laju aliran *feedwater* ke dalam *steam drum*.



Gambar 2.3. P&ID Sistem Pengaturan Level Air *Steam Drum*

2.2.2 Pemodelan *Steam Drum* [11]

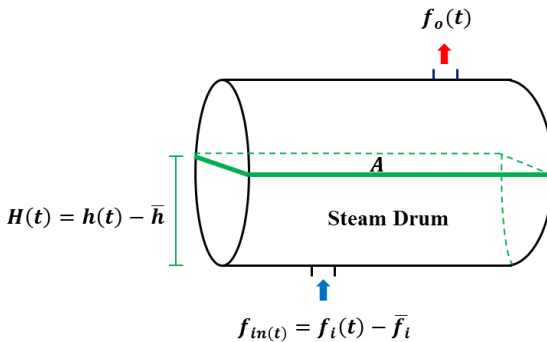
Steam drum merupakan komponen utama *boiler* yang berfungsi untuk memisahkan campuran uap dan air serta mengurangi material padat yang terlarut pada uap. Pengurangan material padat tersebut memiliki tujuan agar sudu turbin tidak terkikis akibat tumbukan material tersebut.

Air dan uap akan memasuki *steam drum* dengan laju dan transfer panas yang merata, hal ini akan menjaga temperatur permukaan bagian dalam *steam drum* relatif sama. Bentuk fisik dari *steam drum* dapat dilihat pada Gambar 2.4. Model matematis *steam drum* diperoleh dari persamaan kesetimbangan massa, sehingga model tersebut dapat dinyatakan seperti pada Persamaan (2.1).



Gambar 2.4. Tampilan Fisik dari *Steam Drum* [12]

Laju massa akumulasi pada suatu sistem adalah pengurangan dari laju aliran massa yang masuk dengan laju aliran massa yang keluar dari sistem. Variabel yang digunakan dalam pemodelan *steam drum* dapat dilihat pada Tabel 2.1. Representasi skema model matematis *steam drum* dapat dilihat pada Gambar 2.5.



Gambar 2.5. Representasi Skema Pemodelan *Steam Drum*

Kesetimbangan massa pada *steam drum*, dapat dinyatakan seperti pada Persamaan (2.1) atau (2.2).

Tabel 2.1. Variabel pada Model Matematis *Steam Drum*

Simbol	Keterangan	Satuan
$\frac{dM}{dt}$	Laju aliran massa akumulasi	kg/s
$f_i(t)$	Laju aliran masuk ke sistem	m ³ /s
$f_o(t)$	Laju aliran keluar dari sistem	m ³ /s
\bar{f}_i	Laju aliran air konstan ke <i>steam drum</i>	m ³ /s
\bar{f}_o	Laju aliran uap keluar dari <i>steam drum</i>	m ³ /s
\bar{h}	Level air konstan didalam <i>steam drum</i>	m
$h(t)$	Level air <i>steam drum</i> saat <i>steady state</i>	m
$H(t)$	Level air didalam <i>steam drum</i>	m
$f_{in}(t)$	Laju masukan air kedalam <i>steam drum</i>	m ³ /s

Laju akumulasi massa = Laju massa masuk - Laju massa keluar (2.1)

$$\frac{dM}{dt} = \rho[f_i(t) - f_o(t)] \quad (2.2)$$

Asumsi bahwa laju aliran air dari pompa *boiler* adalah konstan, maka diperoleh Persamaan (2.3).

$$A \frac{dH(t)}{dt} = f_i(t) - \bar{f}_o \quad (2.3)$$

Ketika kondisi *steady state*, laju aliran air dan uap adalah konstan. Pada kondisi ini, laju aliran air dan uap yang konstan menyebabkan level air *steam drum* konstan. Dengan demikian, laju akumulasi massa pada *steam drum* adalah nol, seperti pada Persamaan (2.4).

$$0 = \bar{f}_i - \bar{f}_o \quad (2.4)$$

Pengurangan antara Persamaan (2.3) dan (2.4), maka diperoleh Persamaan (2.5).

$$A \frac{d[h(t) - \bar{h}]}{dt} = [f_i(t) - \bar{f}_i] - [\bar{f}_o - \bar{f}_o] \quad (2.5)$$

Variabel deviasi level air dan laju aliran *feedwater* dapat dinyatakan seperti pada Persamaan (2.6) dan (2.7) secara berturut-turut.

$$H(t) = h(t) - \bar{h} \quad (2.6)$$

$$f_{in}(t) = f_i(t) - \bar{f}_i \quad (2.7)$$

Substitusi Persamaan (2.6) dan (2.7) ke Persamaan (2.5), sehingga persamaan level air *steam drum* dapat dituliskan menjadi Persamaan (2.8)

$$\frac{dH(t)}{dt} = \frac{f_{in}(t)}{A} \quad (2.8)$$

Fungsi alih sistem tersebut dapat diperoleh dengan melakukan transformasi Laplace pada Persamaan (2.8), sehingga diperoleh fungsi alih *steam drum* seperti pada Persamaan (2.9).

$$G_D(s) = \frac{H(s)}{F_{in}(s)} = \frac{1}{As} \quad (2.9)$$

2.2.3 Pemodelan *Fluid Coupling* [13]

Fluid coupling adalah peralatan hidraulik yang berfungsi untuk menyalurkan gerak rotasi atau torsi. Pada umumnya, pembangkit menggunakan *fluid coupling* untuk mengatur kecepatan pompa *boiler*. Pemodelan *fluid coupling* bertujuan untuk memperoleh nilai penguatan dari peralatan tersebut yang disimbolkan dengan K_V . Nilai penguatan peralatan tersebut dapat diperoleh melalui Persamaan (2.10).

$$K_V = \frac{\text{Rentang daya mekanik yang dihantarkan (kW)}}{\text{Rentang tegangan masukan (V)}} \quad (2.10)$$

2.2.4 Pemodelan Pompa *Boiler* [14]

Jumlah air yang dibutuhkan *steam drum* untuk memenuhi laju aliran uap pada pembangkit diatur oleh pompa *boiler*. Kecepatan aliran pompa *boiler* bergantung pada daya mekanik motor yang diterima oleh fluida pada *fluid coupling* yang menghubungkan antara poros motor dengan poros pompa *boiler*. Pemodelan pompa *boiler* diperlukan untuk mencari besarnya nilai penguatan peralatan tersebut. Nilai penguatan pompa *boiler* dapat dihitung menggunakan Persamaan (2.11).

$$K_p = \frac{\text{Rentang laju aliran } feedwater \text{ ke } steam \text{ drum} (m^3/s)}{\text{Rentang daya mekanik yang dihantarkan} (kW)} \quad (2.11)$$

2.2.5 Pemodelan Pipa [2]

Pipa *boiler* merupakan peralatan yang membawa air setelah dipompa menuju ruang bakar sebelum memasuki *steam drum*. Aliran air pada pipa *boiler* membutuhkan waktu untuk sampai ke *steam drum*. Oleh karena itu, terdapat waktu tunda dalam penyaluran air menuju *steam drum* yang disebabkan oleh panjang pipa. Pemodelan pipa *boiler* bertujuan untuk memperoleh fungsi alih komponen tersebut yang dapat dihitung menggunakan Persamaan (2.12).

$$G_p(s) = e^{-Ls} \quad (2.12)$$

$G_p(s)$ menyatakan waktu tunda yang dibutuhkan oleh *feedwater* untuk mencapai *steam drum*. Waktu tunda dapat dinyatakan dalam fungsi alih melalui pendekatan padè orde pertama seperti pada Persamaan (2.13).

$$G_p(s) = e^{-Ls} \cong \frac{1 - \frac{L}{2}s}{1 + \frac{L}{2}s} \quad (2.13)$$

2.2.6 Pemodelan Level Transmitter [14]

Level transmitter merupakan suatu transduser yang berfungsi untuk mengukur besaran melalui perangkat ukur dan mengubahnya menjadi sinyal standar transmisi. Pemodelan *level transmitter* bertujuan untuk memperoleh nilai penguatan peralatan tersebut yang disimbolkan dengan K_L . Penguatan *level transmitter* dapat dihitung melalui Persamaan (2.14).

$$K_L = \frac{\text{Rentang tegangan keluaran} (V)}{\text{Rentang level air dalam } steam \text{ drum} (m)} \quad (2.14)$$

2.2.7 Pendekatan Padè [15]

. Pendekatan Padè sering digunakan untuk mendekati waktu tunda dengan fungsi alih rasional. Pendekatan Padè dapat dilakukan pada Persamaan (2.12) untuk mendapatkan fungsi alih rasional. Pendekatan

Padè untuk waktu tunda e^{-Ls} dengan nilai r sebagai orde pendekatan dapat dinyatakan dalam Persamaan (2.15).

$$e^{-Ls} \cong \frac{N_r(Ls)}{D_r(Ls)} \quad (2.15)$$

$N_r(Ls)$ dan $D_r(Ls)$ dapat dinyatakan seperti pada Persamaan (2.16) dan (2.17) secara berturut-turut.

$$N_r(Ls) = \sum_{k=0}^r \frac{(2r-k)!}{k!(r-k)!} (-Ls) \quad (2.16)$$

$$D_r(Ls) = \sum_{k=0}^r \frac{(2r-k)!}{k!(r-k)!} (Ls) \quad (2.17)$$

Jika digunakan pendekatan Padè orde pertama maka waktu tunda dapat dinyatakan oleh Persamaan (2.18).

$$e^{-Ls} \cong \frac{1 - \frac{L}{2}s}{1 + \frac{L}{2}s} \quad (2.18)$$

2.3 Kontroler PID [4]

Kontroler PID merupakan salah satu kontroler yang banyak digunakan untuk memperbaiki performa sistem di industri. Struktur kontroler PID yang sederhana menjadi salah satu faktor kontroler tersebut diminati. Parameter kontroler PID adalah penguatan proporsional K_p , waktu integral τ_i , dan waktu derivatif τ_d . Algoritma kontroler PID dapat dijelaskan pada Persamaan (2.19).

$$u(t) = K_p \left(e(t) + \frac{1}{\tau_i} \int_0^t e(\tau) d\tau + \tau_d \frac{de(t)}{dt} \right) \quad (2.19)$$

$u(t)$ merupakan sinyal kontrol dan $e(t)$ adalah sinyal kesalahan. Sinyal kontrol dari kontroler tersebut merupakan penjumlahan dari 3 aksi yaitu, aksi proporsional (P), integral (I) dan derivatif (D). Kontroler PID

memiliki fungsi untuk memperbaiki kesalahan melalui aksi proporsional, menghilangkan kesalahan pada keadaan tunak melalui aksi integral dan mengantisipasi kesalahan akan datang melalui aksi derivatif.

2.3.1 Aksi Proporsional

Aksi proporsional memberikan sinyal kontrol yang proporsional terhadap sinyal kesalahan seperti pada Persamaan (2.20) Penggunaan aksi proporsional saja tidak menghilangkan kesalahan keadaan tunak.

$$u(t) = K_p e(t) \quad (2.20)$$

K_p adalah penguatan proporsional yang dapat mempercepat dan menyebabkan osilasi pada respon sistem apabila nilainya diperbesar. Transformasi Laplace dengan kondisi awal sama dengan nol memberikan fungsi alih aksi proporsional seperti pada Persamaan (2.21)

$$\frac{U(s)}{E(s)} = K_p \quad (2.21)$$

2.3.2 Aksi Integral

Aksi integral berfungsi untuk memastikan bahwa *output* sistem mencapai nilai *setpoint* pada keadaan tunak. Sinyal kesalahan positif akan menyebabkan sinyal kontrol meningkat, sedangkan sinyal kesalahan negatif akan menyebabkan sinyal kontrol menurun pada kontroler dengan aksi integral. Kedua efek aksi integral akan membuat kesalahan pada keadaan tunak bernilai nol. Aksi integral memberikan sinyal kontrol yang proporsional terhadap nilai kesalahan pada waktu sebelumnya seperti pada Persamaan (2.22)

$$u(t) = K_i \left(\frac{1}{\tau_i} \int_0^t e(\tau) d\tau \right) = K_i \int_0^t e(\tau) d\tau \quad (2.22)$$

τ_i merupakan waktu integral dan K_i adalah penguatan integral. Semakin besar nilai waktu integral, maka respon sistem akan semakin lambat untuk mencapai *setpoint*. Nilai waktu integral yang kecil dapat mempercepat dan menyebabkan osilasi pada respon sistem.

Transformasi Laplace dengan kondisi awal sama dengan nol memberikan fungsi alih aksi integral seperti pada Persamaan (2.23)

$$\frac{U(s)}{E(s)} = \frac{K_i}{s} \quad (2.23)$$

2.3.3 Aksi Derivatif

Aksi derivatif bertujuan untuk meningkatkan kestabilan pada sistem *closed loop*. Dinamika proses umumnya menimbulkan waktu tunda pada perubahan sinyal kontrol, sehingga kontroler terlambat dalam memperbaiki sinyal kesalahan. Aksi derivatif memberikan sinyal kontrol yang proporsional terhadap kesalahan akan datang seperti dirumuskan pada Persamaan (2.24)

$$u(t) = K_d \frac{de(t)}{dt} \quad (2.24)$$

K_d adalah penguatan differensial, sehingga fungsi alih aksi derivatif dapat dituliskan seperti Persamaan (2.25)

$$\frac{U(s)}{E(s)} = K_d s \quad (2.25)$$

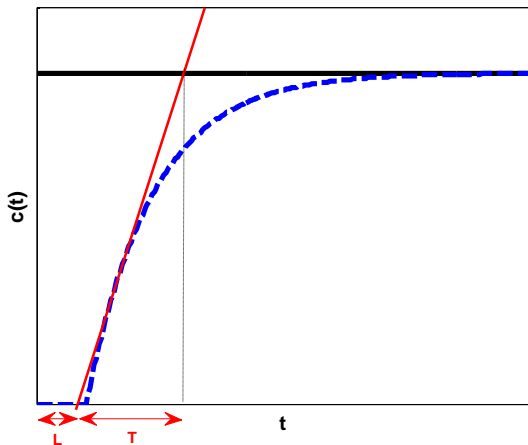
Keuntungan aksi derivatif adalah memiliki *zero* pada titik asal, sehingga sistem menjadi lebih stabil. Aksi derivatif tidak dapat berdiri sendiri karena akan menghasilkan nilai nol saat kesalahan yang terjadi konstan serta akan memperbesar sinyal kontrol. Akibatnya *noise* pada frekuensi tinggi akan diperbesar. Semakin besar nilai waktu derivatif, maka redaman sistem akan membesar. Namun, redaman menurun saat waktu derivatif terlalu besar.

2.4 Aturan *Tuning* Kontroler PID Ziegler-Nichols [16]

Ziegler-Nichols mengusulkan aturan dalam menentukan nilai parameter kontroler PID berdasarkan karakteristik respon sistem. Ziegler-Nichols memiliki dua metode dalam penggunaan aturan *tuning*-nya.

2.4.1 Metode Open Loop

Aturan *tuning* metode *open loop* dilakukan dengan mendapatkan respon sistem terhadap masukan sinyal uji *unit step*. Sistem yang tidak memiliki karakteristik *integrator* atau *pole* dominan *complex-conjugate*, akan menghasilkan respon sistem menyerupai kurva huruf S. Jika respon sistem tidak memiliki bentuk seperti kurva tersebut, maka metode ini tidak dapat diterapkan. Respon sistem menyerupai kurva huruf S seperti pada Gambar 2.6, memiliki dua buah karakteristik, yaitu waktu tunda, L dan waktu konstan, τ yang dapat dilihat. Kedua karakteristik diperoleh dengan menggambar garis singgung yang memotong sumbu waktu dan garis keluaran pada kondisi tunak. Nilai parameter kontroler PID diperoleh dengan mensubstitusikan L dan τ ke dalam Tabel 2.2.



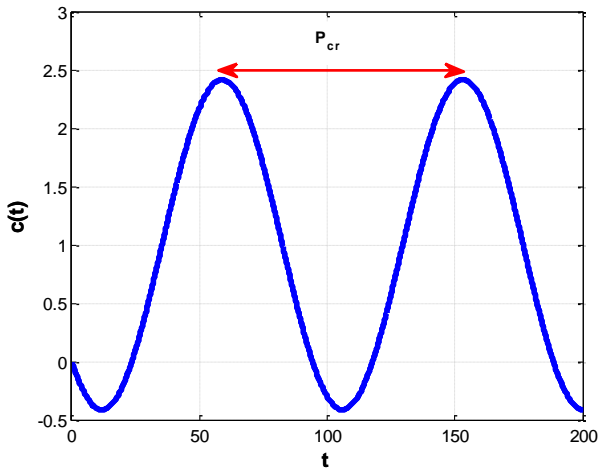
Gambar 2.6. Respon Sistem dengan Bentuk Huruf-S

Tabel 2.2. Aturan *Tuning* Kontroler PID Ziegler-Nichols Metode *Open Loop*

Tipe Kontroler	K_P	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0,9 \frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2 \frac{T}{L}$	$2L$	$0,5L$

2.4.2 Metode *Closed Loop*

Aturan *tuning* Ziegler-Nichols metode *closed loop* dilakukan dengan memperoleh penguatan minimal ketika respon sistem mengalami osilasi permanen. Langkah pertama untuk melakukan *tuning* kontroler PID yaitu mengatur nilai $\tau_i = \infty$ dan $\tau_d = 0$, kemudian tingkatan nilai parameter K_p mulai dari 0 sampai diperoleh nilai penguatan kritis, K_{cr} yaitu ketika respon sistem mengalami osilasi permanen pertama kali. Jika respon sistem tidak mengalami osilasi permanen untuk semua nilai, maka metode ini tidak dapat digunakan. Langkah kedua adalah menentukan periode kritis, P_{cr} , seperti pada Gambar 2.7, yaitu periode ketika sistem mengalami osilasi permanen pertama kali atau dapat disebut periode respon sistem ketika $K_p = K_{cr}$. Aturan *tuning* Ziegler-Nichols metode kedua dapat dilihat pada Tabel 2.3.



Gambar 2.7. Respon Osilasi Permanen dengan Periode P_{cr}

Tabel 2.3. Aturan *Tuning* Kontroler PID Ziegler-Nichols Metode *Closed Loop*

Tipe Kontroler	K_p	T_i	T_d
P	$0,5K_{cr}$	∞	0
PI	$0,45K_{cr}$	$\frac{1}{1,2}P_{cr}$	0
PID	$0,6K_{cr}$	$0,5P_{cr}$	$0,125P_{cr}$

2.5 Indeks Performansi Kesalahan [17]

Kontrol optimal tidak dapat didefinisikan dengan tepat. Solusi yang menurut sebuah masalah adalah kondisi optimal, mungkin bukan nilai yang optimal bagi permasalahan lain. Indeks performansi banyak digunakan oleh kalangan praktisi dan juga akademisi untuk membantu dalam menentukan kualitas sebuah sistem. Indeks performansi sendiri sebenarnya adalah fungsi hubungan dimana beberapa karakteristik sistem seperti kondisi optimal dari sistem didefinisikan. Indeks performansi secara umum dapat dirumuskan pada Persamaan (2.27) dimana J adalah indeks performansi dan e adalah kesalahan.

$$J = \int_0^{\infty} f(e)dt \quad (2.26)$$

Nilai indeks performansi ini mengindikasikan seberapa bagus performansi dari suatu sistem. Pada penjelasan selanjutnya akan dibahas mengenai beberapa indeks performansi untuk kriteria kesalahan yang biasa digunakan dalam perancangan kontroler.

2.5.1 *Integral Square Error (ISE)*

Indeks performansi ISE ditunjukkan pada Persamaan (2.28). Indeks performansi ini berhubungan dengan *Mean Square Error* (MSE).

$$J = \int_0^{\infty} e^2(t)dt \quad (2.27)$$

ISE selain dapat digunakan untuk mendesain sebuah kontroler, juga dapat digunakan sebagai kriteria analisa mengenai seberapa baik sebuah sistem dengan kontroler dalam mengatasi pembebanan.

2.5.2 *Integral of Absolute Value of Error (IAE)*

Indeks performansi IAE ditunjukkan pada Persamaan (2.29). IAE memiliki struktur kriteria yang paling mudah diaplikasikan dibandingkan dengan kriteria kesalahan lainnya, karena secara matematis indeks performansi tersebut mudah dioperasikan.

$$J = \int_0^{\infty} |e(t)| dt \quad (2.28)$$

Kriteria ini dapat menghilangkan kesalahan yang besar dan kesalahan yang kecil jika dibandingkan dengan ISE. Sistem optimal berdasarkan kriteria ini mempunyai nilai redaman dan fasa transien yang sangat baik.

2.5.3 *Integral of Time Multiplied by Absolute Value of Error (ITAE)*

Indeks performansi dari ITAE ditunjukkan pada Persamaan 2.30. Bobot pengali waktu pada ITAE membuatnya lebih cepat untuk meminimalkan kesalahan dibandingkan IAE.

$$J = \int_0^{\infty} t |e(t)| dt \quad (2.29)$$

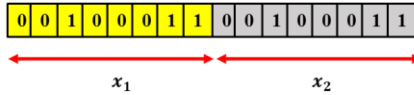
ITAE memiliki nilai kesalahan awal yang besar dan kemudian kesalahan tersebut menghilang secara perlahan.

2.6 *Genetic Algorithm (GA)* [18]

GA merupakan metode pencarian acak yang meniru mekanisme evolusi pada makhluk hidup. GA mencari solusi pada kumpulan data melalui prinsip ketahanan hidup. Algoritma ini membentuk populasi yang terdiri dari beberapa individu untuk menyatakan kemungkinan solusi. Setiap individu dievaluasi ketahanannya melalui fungsi *fitness*. Semakin tinggi nilai ketahanan individu, maka memiliki peluang lebih besar terpilih mengikuti proses reproduksi.

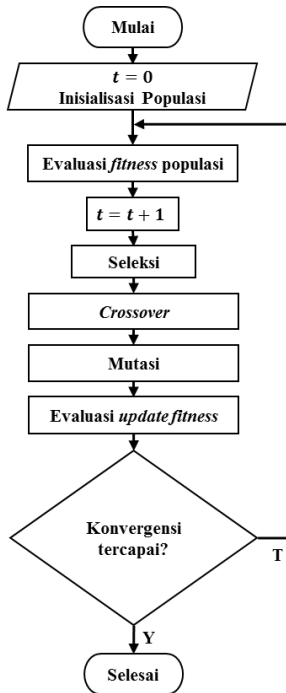
Solusi permasalahan direpresentasikan oleh pengkodean individu dalam kromosom yang terdiri dari beberapa karakter, sehingga nilai kromosom. Representasi GA paling banyak menggunakan pengkodean bilangan biner, meskipun terdapat cara representasi lain seperti *ternary*, bilangan bulat, atau bilangan riil. Contohnya, permasalahan ditentukan memiliki solusi optimal berupa variabel x_1 dan x_2 yang dipetakan pada kromosom seperti pada Gambar 2.8. Proses reproduksi meliputi *crossover* dan mutasi. Setelah melalui proses reproduksi, maka terbentuk generasi

baru. Jika solusi masih belum optimal, maka akan melalui proses seleksi dan reproduksi kembali.



Gambar 2.8. Representasi Kromosom pada GA

Dengan demikian, populasi akan semakin mendekati solusi yang optimal. Mekanisme GA dapat dijelaskan melalui Gambar 2.9.

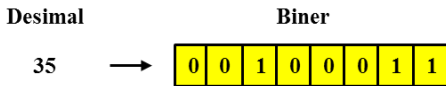


Gambar 2.9. Diagram Alir GA

2.6.1 Representasi dan Inisialisasi Populasi

GA tidak mencari solusi pada suatu data atau nilai, melainkan dari sekumpulan data atau nilai yang disebut populasi. Pada umumnya,

populasi pada GA terdiri dari 30-100 individu, namun khusus untuk mikro GA terdiri ± 10 individu untuk aplikasi yang membutuhkan *real-time*. Representasi kromosom pada individu di dalam GA ada beberapa tipe, diantaranya pengkodean berupa bilangan biner, integer, dan riil. Contoh representasi solusi bilangan biner dapat dilihat pada Gambar 2.10.



Gambar 2.10. Representasi Kromosom dengan Bilangan Biner

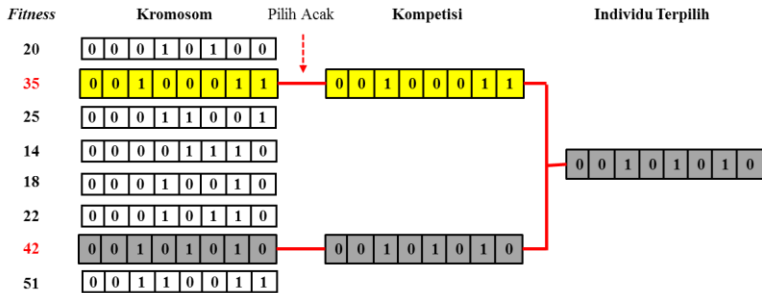
2.6.2 Fungsi Objektif dan Fungsi *Fitness*

Fungsi objektif, J , digunakan untuk mengukur performa atau ketahanan individu sebagai solusi suatu permasalahan. Pada kasus permasalahan minimisasi, individu yang sesuai memiliki nilai fungsi objektif yang paling rendah. Nilai ketahanan yang diperoleh dari fungsi objektif digunakan pada tahap pertengahan untuk menentukan nilai ketahanan relatif individu. Fungsi *fitness*, $f = g(J)$, pada umumnya digunakan untuk merubah nilai fungsi objektif untuk mengukur nilai ketahanan relatif individu. Fungsi *fitness* digunakan untuk mengubah nilai fungsi objektif menjadi bilangan non-negatif.

2.6.3 Seleksi

Operator seleksi merupakan salah satu komponen yang memegang peran penting dalam mendesain GA. Seleksi berfungsi untuk menentukan individu yang akan berperan untuk membentuk populasi baru. Tujuan dari seleksi adalah memastikan bahwa individu yang terpilih mengikuti proses reproduksi memiliki nilai *fitness* yang tinggi. Terdapat beberapa tipe operator seleksi yang populer, yaitu *roulette*, *tournament*, dan *ranking*. Salah satu ilustrasi dari tipe operator seleksi, yaitu *tournament* dapat dilihat pada Gambar 2.11.

Tournament merupakan tipe operator seleksi yang paling populer dan termasuk skema seleksi yang efisien, karena mekanismenya sederhana dan ketika diimplementasikan memberikan hasil yang memuaskan. Prinsip *tournament* adalah memilih individu secara acak dari populasi, kemudian kompetisi dilakukan berupa perbandingan nilai *fitness* misalnya seperti ilustrasi pada Gambar 2.11. Individu dengan nilai *fitness* lebih tinggi akan terpilih untuk mengikuti proses reproduksi.

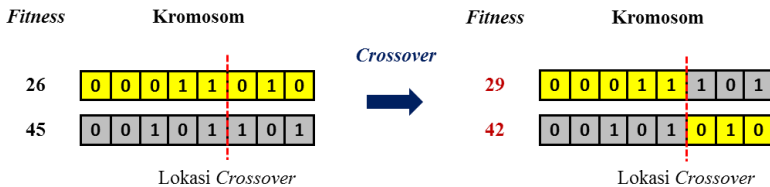


Gambar 2.11. Mekanisme Seleksi Tipe *Tournament*

2.6.4 Crossover

Operator *crossover* meniru mekanisme alami pindah silang yang terjadi pada suatu kromosom untuk diaplikasikan secara acak pada individu yang terpilih dari hasil operator seleksi. Prinsip dari *crossover* adalah menukar sebagian informasi dari individu dengan individu pasangannya. Operator *crossover* memiliki beberapa tipe diantaranya adalah *single-point*, *multipoint*, *uniform*, *intermediate recombination*, dan *line recombination*.

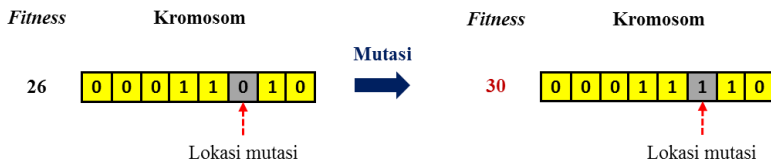
Single crossover merupakan mekanisme *crossover* paling sederhana dengan melakukan pemilihan individu secara acak dari populasi hasil seleksi untuk menghasilkan individu baru yang memiliki mekanisme seperti pada Gambar 2.12. Kemudian informasi pada individu tersebut dipisahkan menjadi sisi bagian kiri dan kanan, dengan lokasi terjadinya pemisahan dipilih secara acak. Setelah itu, informasi pada individu tersebut saling bertukar pada posisi yang sama, sehingga terbentuk individu baru. Ketika individu telah mengalami *crossover*, maka nilai *fitness* dari masing-masing individu mengalami perubahan.



Gambar 2.12. Mekanisme *Single-point Crossover*

2.6.5 Mutasi

Mutasi pada evolusi makhluk hidup merupakan proses pergantian satu alel suatu gen dengan alel gen lain untuk menghasilkan struktur genetika yang baru. Operator mutasi pada GA dapat menghasilkan individu baru yang tidak mungkin dapat diperoleh dari operator seleksi atau *crossover*. Mutasi berperan untuk menjaga keberagaman dalam populasi dengan cara mengubah informasi pada suatu individu. Tipe operator mutasi pada GA ada beberapa macam, seperti *insert mutation*, *uniform mutation*, *flip mutation* dan lain sebagainya.

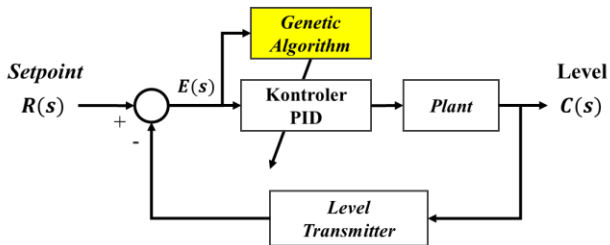


Gambar 2.13. Mekanisme *Flip Mutation*

Mekanisme mutasi tipe *flip* adalah dengan memilih secara acak lokasi terjadinya mutasi pada kromosom, kemudian informasi genetika anantara kedua induk dalam hal ini berupa nilai bit pada kromosom tersebut diubah menjadi lawan bilangan binernya (0 menjadi 1 dan 1 menjadi 0).

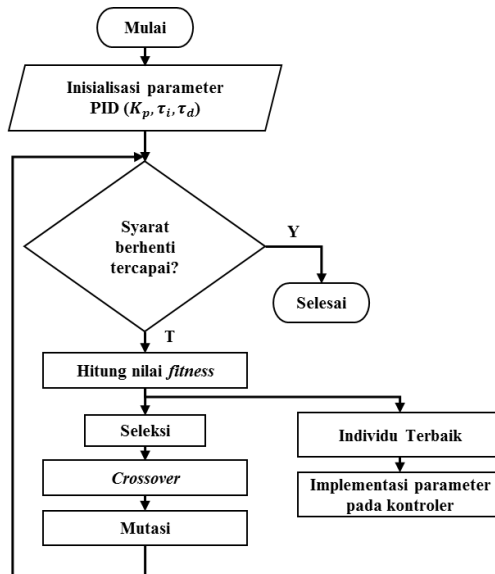
2.7 Kontroler PID-Genetic Algorithm (PID-GA) [7]

Kontroler PID-GA merupakan kontroler PID dengan penambahan mekanisme optimisasi pada *tuning* parameternya berdasarkan mekanisme GA. Algoritma tersebut digunakan sebagai metode optimisasi untuk mencari nilai parameter optimal berdasarkan fungsi objektif yang ditentukan. Pada kasus ini, fungsi objektifnya adalah meminimalkan kesalahan antara nilai setpoint dengan keluaran aktual. Nilai parameter kontroler pada kondisi awal ditentukan secara acak, lalu digunakan sebagai parameter awal kontroler PID.



Gambar 2.14. Diagram Blok Sistem Pengaturan dengan Kontroler PID-GA

Sistem dioperasikan menggunakan parameter awal dan dilakukan evaluasi terhadap fungsi *fitness* untuk setiap pasangan parameter.



Gambar 2.15. Diagram Alir *Online Tuning* Kontroler PID-GA

Pasangan dengan nilai *fitness* yang tinggi akan terpilih pada proses reproduksi yang meliputi seleksi, *crossover*, dan mutasi. Pasangan parameter yang telah melalui proses reproduksi dievaluasi kembali nilai *fitness*-nya. Setelah itu, mekanisme akan ditentukan oleh syarat

konvergensi yang menyatakan bahwa pasangan parameter tersebut adalah parameter yang optimal. Jika syarat konvergensi terpenuhi, maka nilai parameter tersebut digunakan sebagai parameter kontroler PID-GA. Namun jika belum terpenuhi, maka dilakukan mekanisme *update* parameter kontroler melalui proses reproduksi. Keseluruhan mekanisme kontroler PID-GA dapat digambarkan menggunakan diagram alir seperti pada Gambar 2.15.

BAB 3

SIMULATOR PLTU DAN *FEEDWATER SYSTEM*

Simulator yang dirancang pada penelitian ini fokus pada HMI *feedwater systems*. Arsitektur simulator beserta jaringan komunikasi dan HMI yang digunakan dibahas pada bab ini. Pemodelan sistem dirancang berdasarkan nilai parameter yang ditentukan.

3.1 Perancangan Simulator PLTU

Simulator adalah peralatan berupa perangkat keras atau lunak yang dirancang untuk meniru operasi atau pengaturan pada kendaraan, pesawat, atau sistem kompleks lainnya. Simulator umumnya digunakan sebagai media pembelajaran atau pelatihan dalam mengoperasikan suatu peralatan. Pembelajaran atau pelatihan tersebut bertujuan untuk mengenalkan dan meningkatkan kemampuan operator tentang suatu proses yang ditangani. Dengan demikian, operator memiliki bekal yang cukup untuk mengoperasikan suatu sistem dengan benar serta mampu mengatasi permasalahan yang umum terjadi pada sistem tersebut.

Simulator pada penelitian ini berupa perangkat lunak yang dirancang menyerupai sistem riil pada PLTU di PT. PJB UP 1 dan 2 Paiton terutama pada segi tampilan *Human Machine Interface* (HMI). Proses PLTU pada perusahaan tersebut dikelompokkan menjadi 16 sub-proses HMI, yaitu:

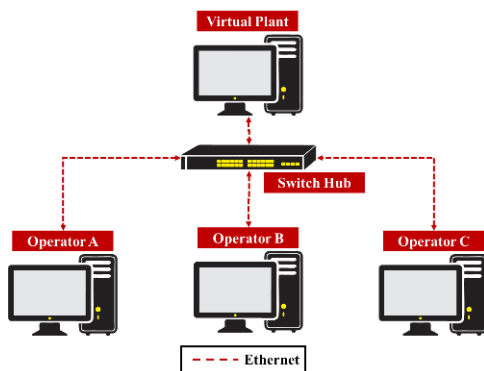
- a) *Unit Master*
- b) *Boiler Firing*
- c) *Burner Coal Feed*
- d) *Fuel Oil & Gas*
- e) *Draft Plant*
- f) *Main & Reheat Steam*
- g) *Boiler Auxiliaries*
- h) *Plant Electrical*
- i) *Turbine Supervisory*
- j) *Turbine Steam*
- k) *Turbine Auxiliaries*
- l) *DEHC Overview*
- m) *Feedwater System*
- n) *Condensate System*
- o) *Cooling Water Systems*
- p) *Misc. Systems*

Penelitian ini fokus pada perancangan HMI *Feedwater Systems* yaitu salah satu sistem pada PLTU di PT. PJB UP 1 dan 2 Paiton yang besaran didalamnya membutuhkan pengawasan operator. Komponen utama pada sistem tersebut adalah *boiler feed pump* (BFP), *high pressure heater* (HPH), *steam drum*, dan *deaerator*. *Feedwater systems* merupakan sistem yang berisi komponen untuk mengolah air pada proses pembentukan uap. Pada HMI tersebut terdapat pengawasan tentang level air *steam drum*. Sistem tersebut memiliki sub-proses HMI didalamnya, antara lain:

- a) *Feedwater Systems Overview*
- b) *Boiler Feed Pumps (systems)*
- c) *Boiler Feed Pumps (components)*
- d) *BFP Permissives*
- e) *BFP Sequences*
- f) *High Pressure Heaters*

3.2 Arsitektur Simulator PLTU

Simulator PLTU yang dirancang terdiri dari komponen virtual plant, HMI operator, dan jaringan komunikasi. *Virtual plant* merupakan perangkat lunak yang berisi model matematis dan algoritma kontrol suatu sistem. Perangkat lunak tersebut dilengkapi dengan fasilitas untuk mengubah masukan atau parameter kontroler. HMI operator merupakan perangkat lunak yang terdiri dari tampilan proses beserta nilai besaran yang terukur pada riil sistem.



Gambar 3.1. Arsitektur Simulator PLTU

Perangkat tersebut dapat mempermudah operator dalam mengamati besaran pada plant ketika beroperasi. Komponen ketiga pada simulator PLTU yang dirancang adalah jaringan komunikasi. Pada penelitian ini, simulator menggunakan jaringan komunikasi dengan media kabel *ethernet* dan perangkat *switch hub*, sehingga arsitektur simulator tersebut dapat dilihat pada Gambar 3.1.

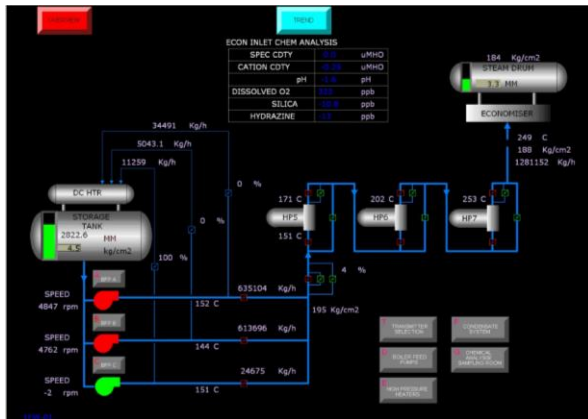
3.2.1 Jaringan Komunikasi

Pertukaran data antara komputer *virtual plant* dengan komputer operator menggunakan teknologi *network-published shared variables* yang dikembangkan oleh perusahaan National Instruments. Teknologi tersebut mengirimkan data ke jaringan melalui komponen perangkat lunak yang disebut *Shared Variable Engine* (SVE). Pertukaran data pada suatu jaringan membutuhkan adanya aturan komunikasi yang disebut protokol jaringan. Protokol komunikasi yang digunakan pada simulator PLTU ini adalah *NI Publish Subscribe Protocol* (NI-PSP).

Data yang ingin dikirim pada jaringan komunikasi diletakkan pada variabel yang dibuat pada *project*. Data tersebut dihubungkan dengan SVE. Jika data tersebut telah disimpan dan dikirim ke jaringan, maka masing-masing komputer yang terhubung ke jaringan dapat saling bertukar data. Dengan demikian, data yang dihasilkan pada komputer *virtual plant* dapat diakses oleh komputer operator.

3.2.2 HMI Feedwater Systems Overview

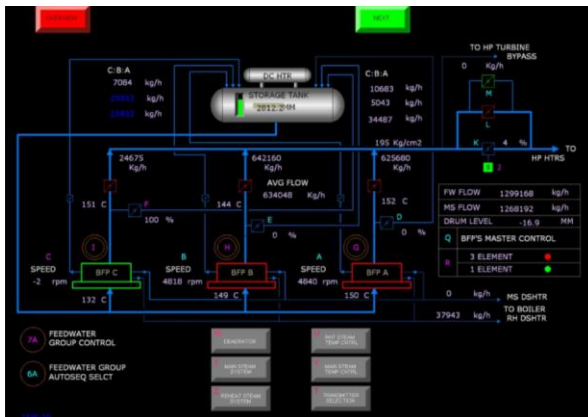
HMI *feedwater system overview* menampilkan informasi mengenai besaran-besaran utama yang perlu diawasi oleh operator pada komponen sistem tersebut ketika pembangkit beroperasi. Level air *steam drum* merupakan salah satu besaran yang menjadi perhatian pada HMI ini. Level air tersebut dipengaruhi oleh laju aliran *feedwater* yang masuk dan laju aliran uap yang keluar dari drum. Tampilan HMI *feedwater systems overview* dapat dilihat pada Gambar 3.2.



Gambar 3.2. Tampilan HMI *Feedwater Systems Overview*

3.2.3 HMI *Boiler Feed Pumps (Systems)*

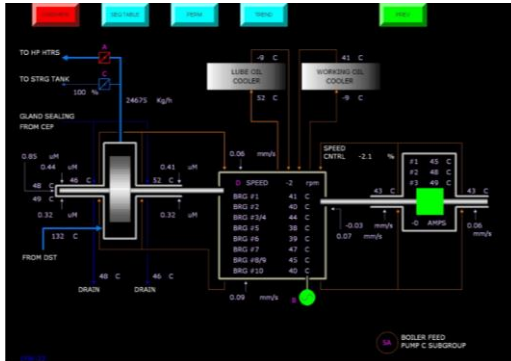
HMI ini berisi besaran yang terdapat pada sistem pengisian *feedwater* yang berasal dari tangki *deaerator* lalu dipompa oleh BFP menuju *steam drum* dengan melalui HPH dan *economizer* terlebih dahulu. HMI *boiler feed pumps (systems)* dapat dilihat pada Gambar 3.3



Gambar 3.3. Tampilan HMI *Boiler Feed Pumps (Systems)*

3.2.4 HMI Boiler Feed Pumps (Components)

HMI ini menampilkan informasi tentang besaran temperatur BFP seperti pada komponen *bearing*. Tampilan HMI BFP *components* dapat dilihat pada Gambar 3.4.



Gambar 3.4. Tampilan HMI *Boiler Feed Pumps (Components)*

3.2.5 HMI BFP Sequences

BFP merupakan pompa yang digunakan untuk mengalirkan *feedwater* menuju *steam drum*. Proses penyalan pompa tersebut harus memenuhi prosedur yang ditentukan. Langkah penyalan BFP dapat dilihat oleh operator pada tampilan HMI seperti pada Gambar 3.5.



Gambar 3.5. Tampilan HMI *BFP Sequences*

3.2.6 HMI BFP Permissives

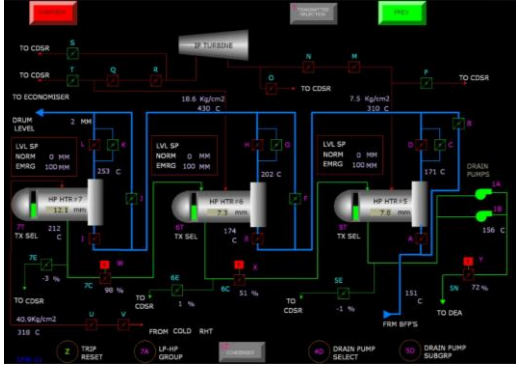
Informasi tentang persyaratan operasi normal BFP dapat dilihat oleh operator pada tampilan HMI seperti pada Gambar 3.6. HMI ini memberikan status operasi dari peralatan proteksi pada BFP.



Gambar 3.6. Tampilan HMI BFP Permissives

3.2.7 HMI High Pressure Heaters (HPH)

HPH pada PLTU berfungsi untuk meningkatkan efisiensi boiler dengan cara meningkatkan temperatur air sebelum memasuki economizer dan steam drum. Besaran yang membutuhkan pengawasan dari operator adalah level air didalam HPH. Tampilan HMI untuk melakukan pengawasan dan pengaturan pada HPH dapat dilihat pada Gambar 3.7.

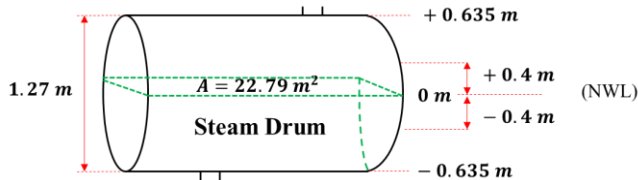


Gambar 3.7. Tampilan HMI High Pressure Heater (HPH)

3.3 Pemodelan Sistem Pengaturan Level Air *Steam Drum*

Sistem pengaturan level air *steam drum* melibatkan beberapa komponen, seperti *steam drum*, *voith coupling*, pompa boiler, pipa, dan *level transmitter*. Batas normal level air pada *steam drum* ketika beroperasi normal yaitu berada pada setengah dari ketinggian drum, yang disebut sebagai titik *Normally Water Level* (NWL) seperti diilustrasikan pada Gambar 3.8. Rentang toleransi level air *steam drum* ketika beroperasi adalah ± 40 cm, apabila diluar batas tersebut maka proteksi *steam drum* bekerja untuk menghentikan proses pada boiler.

Proses perubahan level air pada *steam drum* berawal dari *feedwater* dialirkan oleh pompa boiler yang terhubung dengan *fluid coupling* melewati pipa-pipa boiler menuju *steam drum*. Level air di dalam *steam drum* diukur oleh *level transmitter*, kemudian level aktual dibandingkan dengan *setpoint* untuk memperoleh kesalahan dalam sistem tersebut. Uap yang telah dipisahkan oleh *steam drum* dialirkan menuju pemanas tingkat lanjut, sehingga level air di dalam *steam drum* menurun. Jika terjadi penurunan level air, maka pompa akan menambahkan *feedwater* ke dalam *steam drum*, sehingga level air dapat memenuhi *setpoint*.



Gambar 3.8. Representasi Skema *Steam Drum*

Pemodelan sistem pengaturan level air *steam drum* dapat diperoleh dengan mencari model matematis dari setiap komponen seperti pada diagram blok Gambar 3.9. Pada penelitian ini, *steam drum* yang digunakan merujuk pada Tabel 3.1. Tipe yang digunakan adalah horizontal *steam drum* berbentuk tabung dengan ukuran panjang 16,15 m dan diameter sebesar 22,79 m². Nilai luas penampang melintang *steam drum* adalah sebesar 22,79 m². Dengan demikian, fungsi alihnya dapat dinyatakan oleh Persamaan (3.1).

Tabel 3.1. Parameter pada Model Matematis *Steam Drum*

Simbol	Keterangan	Nilai	Satuan
ρ	Massa jenis cairan	833,69	kg/m^3
A	Luas penampang melintang <i>steam drum</i>	22,79	m^2
V	Volume <i>steam drum</i>	20,46	m^3

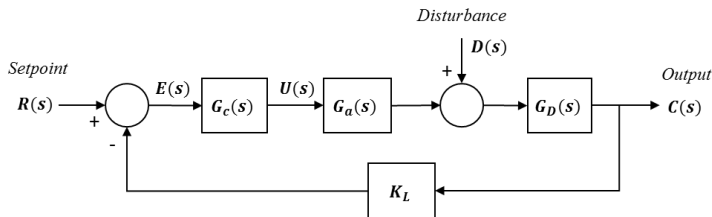
$$G_D(s) = \frac{1}{22,79s} \quad (3.1)$$

Daya mekanik yang mampu dihantarkan oleh *fluid coupling* 0 – 7100 kW, sedangkan rentang tegangan yang digunakan adalah sebesar 0 – 10 V. Oleh karena itu, nilai penguatan *fluid coupling*, K_V dapat diperoleh seperti pada Persamaan (3.2).

$$K_V = \frac{7100-0}{10-0} = 710 \text{ kW/V} \quad (3.2)$$

Rentang nilai laju aliran *feedwater* adalah 0 – 0,2186 m^3/s , sehingga nilai penguatan pompa *boiler* dinyatakan oleh Persamaan (3.3)

$$K_P = \frac{0.2186-0}{7100-0} = 3,1 \times 10^{-5} \text{ m}^3 \text{ kW/s} \quad (3.3)$$



Gambar 3.9. Diagram Blok Sistem Pengaturan *Closed Loop*

Waktu tunda yang dibutuhkan oleh *feedwater* untuk mencapai *steam drum* adalah 30 detik. Dengan demikian, model matematis pipa *boiler* dapat diperoleh seperti pada Persamaan (3.4).

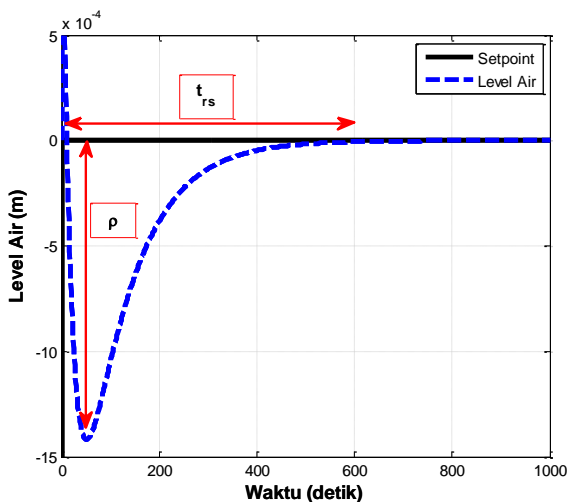
$$G_P(s) = \frac{1-15s}{1+15s} \quad (3.4)$$

Rentang tegangan keluaran *level transmitter* diperoleh sebesar 0 – 10 V, sedangkan rentang level air di dalam *steam drum* adalah sebesar 0 – 1,27 m. Berdasarkan data tersebut, model *level transmitter* dapat diperoleh dari nilai penguatannya seperti pada Persamaan (3.5).

$$K_L = \frac{10-0}{1,27-0} = 7,874V/m \quad (3.5)$$

3.4 Spesifikasi Respon

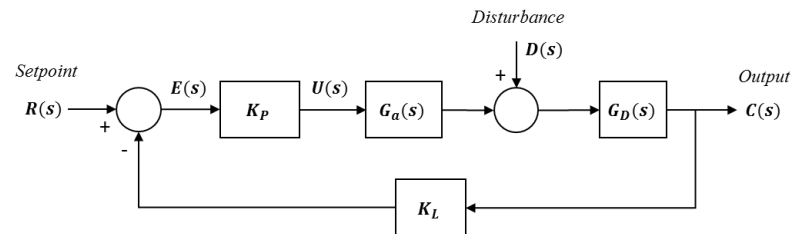
Sistem pengaturan level air *steam drum* diharapkan mampu menghilangkan efek gangguan pada sistem dan menjaga nilai keluaran sistem sedekat mungkin dengan nilai *setpoint*. Spesifikasi sistem pertama yang diamati adalah *perturbation peak*, ρ , yaitu nilai puncak respon gangguan diukur dari nilai keluaran pada waktu tunak. Spesifikasi sistem kedua adalah *recovery time*, t_{rs} , yaitu waktu yang dibutuhkan respon gangguan untuk mencapai kondisi tunak dan berada pada batas kesalahan yang ditentukan seperti pada Gambar 3.10. Spesifikasi respon yang diinginkan yaitu memiliki *perturbation peak* kurang dari 0.25 m dan *recovery time* kurang dari 350 detik.



Gambar 3.10. Spesifikasi Respon Level Air *Steam Drum*

3.5 Desain Kontroler PID Ziegler-Nichols (PID-ZN)

Respon sistem *open loop* memiliki karakteristik integrator, sehingga aturan *tuning* kontroler PID-ZN metode pertama tidak dapat digunakan. Oleh karena itu, sistem pengaturan level air *steam drum* nilai parameter kontrolernya diperoleh menggunakan metode kedua. Berdasarkan hasil pengujian nilai penguatan proporsional, K_p yang menyebabkan respon sistem mengalami osilasi permanen untuk pertama kali adalah $K_p = K_{cr} = 8,78$. Periode kritis, P_{cr} yaitu ketika sistem mengalami osilasi permanen pertama kali adalah sebesar $P_{cr} = 95$ detik. Diagram blok sistem untuk melakukan *tuning* menggunakan aturan ZN dapat dilihat pada Gambar 3.11. Hasil respon sistem ketika mengalami osilasi permanen untuk pertama kali adalah seperti pada Gambar 3.12.

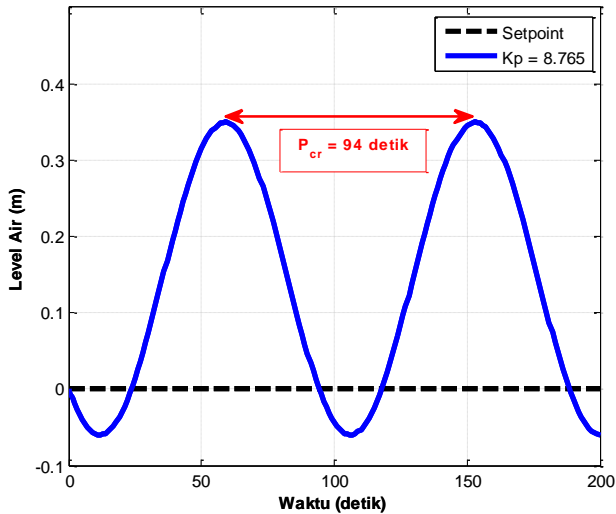


Gambar 3.11. Diagram Blok *Tuning* PID-ZN Metode *Closed-Loop*

Berdasarkan dua konstanta yang diperoleh, yaitu K_{cr} dan P_{cr} maka nilai parameter kontroler P, PI, dan PID dapat diperoleh dengan mensubstitusikan kedua konstanta pada Tabel 2.2. Hasil *tuning* parameter kontroler PID menggunakan aturan Ziegler-Nichols metode *closed-loop* dapat dilihat pada Tabel 3.2.

Tabel 3.2. Nilai Parameter Kontroler PID-ZN Metode *Closed Loop*

Tipe Kontroler	K_p	T_i	T_d
P	4,383	∞	0
PI	3,94	78,33	0
PID	5,26	47	11,75



Gambar 3.12. Respon Sistem dengan $K_p = 8,78$

3.6 Desain Kontroler PID-GA

Parameter kontroler PID pada penelitian ini direpresentasikan dalam bentuk bilangan biner seperti pada Gambar 3.13. Ukuran individu yang digunakan adalah sebesar 30 bit, dengan setiap parameter memiliki ukuran 10 bit. Satu buah calon solusi optimal terdiri dari tiga pasang nilai yaitu K_p , τ_i , dan τ_d , sehingga satu buah individu pada GA dapat diilustrasikan seperti pada Gambar 3.14. Tipe operator GA yang digunakan yaitu, seleksi *tournament*, *single-point crossover*, dan *flip mutation*. Adapun nilai probabilitas terjadinya crossover dan mutasi berturut-turut adalah 0,6 dan 0,02.

K_p	τ_i	τ_d
-------	----------	----------

Gambar 3.13. Representasi Individu GA

Fungsi objektif yang digunakan pada penelitian ini merupakan kombinasi dari tiga jenis indeks performansi kesalahan, yaitu ISE, IAE, dan ITAE. Tingkat peranan indeks performansi tersebut diwakili oleh faktor pembobot w . Keterlibatan efek indeks performansi ISE, IAE, dan

ITAE dinotasikan sebagai $w_1 = 0,5$; $w_2 = 0,2$; dan $w_3 = 0,3$ secara berturut-turut. Tujuan dari fungsi objektif ini adalah meminimalkan kesalahan sistem, sehingga diperoleh nilai *fitness* individu semakin besar ketika solusi semakin sesuai dengan persoalan. Pada penelitian ini, fungsi objektif dan fungsi *fitness* yang digunakan secara berurutan dinyatakan pada Persamaan (3.6) dan (3.7) [7].

$$J = w_1 (ISE) + w_2 (IAE) + w_3 (ITAE) \quad (3.6)$$

$$f = \frac{1}{1 + J} \quad (3.7)$$

Representasi parameter kontroler PID yang digunakan pada GA adalah bilangan biner. Parameter kontroler yang digunakan pada sistem merupakan bilangan riil. Dengan demikian, mekanisme kontroler PID-GA membutuhkan adanya proses untuk mengubah representasi biner menjadi riil agar dapat diimplementasikan pada sistem. Persamaan yang digunakan untuk merubah representasi biner menjadi riil dapat dilihat pada Persamaan (3.8) [19].

$$x_i = x_{\min} + \frac{x_{\max} - x_{\min}}{2^L - 1} y_i \quad (i = 1, 2, 3) \quad (3.8)$$

BAB 4

PENGUJIAN DAN ANALISIS METODE KONTROL PID DAN PID-GA

Pengujian dan analisis pada bab ini dilakukan untuk mengetahui performa sistem. Penambahan umpan balik mampu memperbaiki kestabilan sistem, namun gangguan pada sistem menyebabkan respon tidak kembali menuju *setpoint*. Pengujian sistem dengan kontroler PID dilakukan untuk menganalisis performa sistem ketika menerima gangguan, kemudian dilakukan perbandingan performa sistem dengan kontroler PID konvensional dan kontroler PID-GA.

4.1 Deskripsi Beban pada Pengujian Metode Kontrol

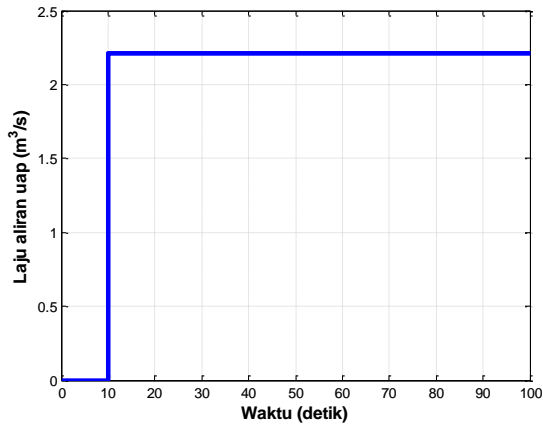
Beban yang menjadi gangguan pada sistem pengaturan level air *steam drum* berupa laju aliran uap yang keluar dari drum. Beban pada pembangkit selama beroperasi tidak selamanya tetap, ada saatnya berubah sesuai permintaan *dispatcher* atau Penyaluran dan Pusat Pengatur Beban (P3B). Pada penelitian ini besarnya beban dikelompokkan menjadi tiga kelompok, yaitu minimal, nominal, dan maksimal. Beban yang digunakan dalam pengujian dan analisis metode kontrol dituliskan pada Tabel 4.1. Jenis beban yang digunakan pada penelitian ini untuk merepresentasikan sistem riilnya adalah beban tetap, beban acak, serta beban dengan perubahan kriteria.

Tabel 4.1. Kriteria Beban Pembangkit dan Laju Aliran Uap

Beban	Daya Pembangkit (MW)	Laju aliran uap keluar dari <i>steam drum</i> (m^3/s)
Minimal	297,48	2,21
Nominal	349,42	2,63
Maksimal	399,16	3,10

4.1.1 Beban Tetap

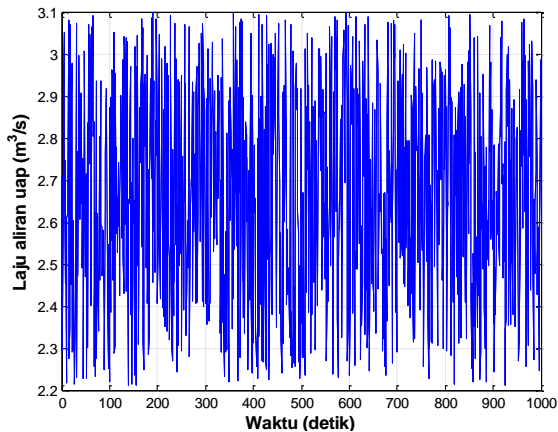
Beban tetap digunakan untuk menganalisis respon sistem ketika pada sistem riil *dispatcher* atau P3B meminta beban yang tetap pada selang waktu tertentu. Hal tersebut menyebabkan laju aliran uap yang keluar dari drum berubah. Perubahan beban direpresentasikan berupa sinyal *step* dengan variasi nilai akhir sesuai kriteria beban, yaitu 2,21; 2,63; dan 3,10 seperti pada Gambar 4.1.



Gambar 4.1. Representasi Perubahan Beban Tetap

4.1.2 Beban Acak

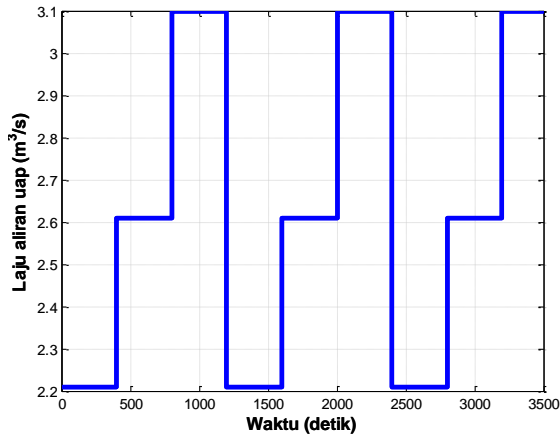
Beban acak digunakan untuk menganalisis respon sistem ketika beban pada sistem riil bervariasi pada rentang tertentu. Perubahan laju aliran uap pada kondisi ideal memiliki nilai acak. Oleh karena itu, untuk menganalisis performa sistem dengan gangguan acak, maka dilakukan pengujian dengan beban acak yang diberikan seperti pada Gambar 4.2.



Gambar 4.2. Representasi Perubahan Beban Acak

4.1.3 Beban dengan Perubahan Kriteria

Beban dengan perubahan kriteria pada pengujian ini diberikan untuk merepresentasikan beban pada pembangkit selama beroperasi mengalami perubahan nilai pada kondisi minimal, nominal, dan maksimal. Kondisi perubahan tersebut terjadi akibat konsumsi energi listrik konsumen tidak konstan, misalkan pada dini hari konsumsi energi menurun, sehingga beban pembangkit minimal. Ketika mulai memasuki jam kerja, kebutuhan energi meningkat, sehingga pembangkit menerima beban nomina. Menjelang sore hari, kebutuhan energi naik kembali akibat penggunaan penerangan dan perangkat lainnya, sehingga pembangkit menerima beban maksimal. Hal ini berulang kembali pada hari berikutnya. Dengan demikian, beban tersebut dapat direpresentasikan dengan perubahan nilai akhir sinyal *step* seperti pada Gambar 4.3.



Gambar 4.3. Representasi Beban dengan Perubahan Kriteria

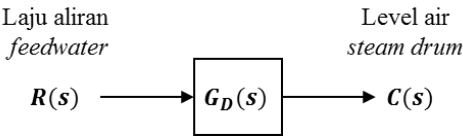
4.2 Pengujian Sistem secara *Open Loop*

Model matematis *steam drum* telah diperoleh dan dinyatakan pada Persamaan (3.1), sehingga dapat dituliskan kembali pada Persamaan (4.1)

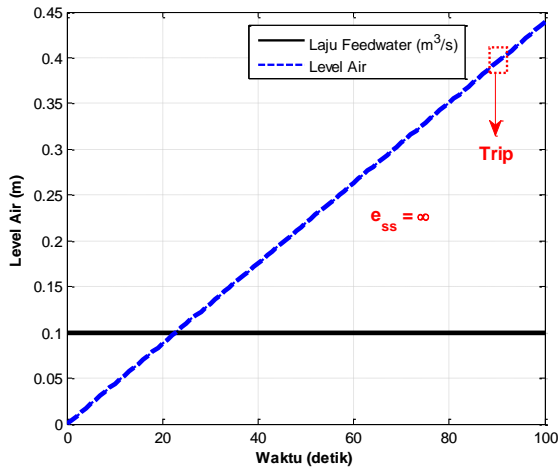
$$G_D(s) = \frac{1}{22,79s} \quad (4.1)$$

Karakteristik respon level air *steam drum* diperoleh melalui pengujian secara *open loop* dengan diagram blok seperti pada Gambar 4.3. Respon

hasil simulasi level air *steam drum* terhadap laju aliran *feedwater* sebesar $0,1 \text{ m}^3/\text{s}$ dapat dilihat pada Gambar 4.4. Pada simulasi tersebut masukan berupa laju aliran *feedwater* direpresentasikan oleh sinyal uji *unit step* dengan nilai akhir 0,1 dan kondisi awal level air berada pada level 0 m.



Gambar 4.4. Diagram Blok Pengujian Sistem *Open Loop*



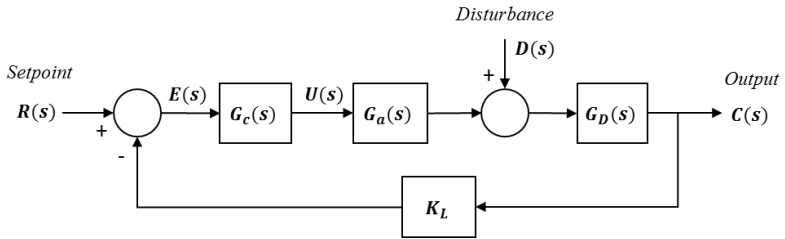
Gambar 4.5. Respon Level dengan Masukan Sinyal Uji *Step*

Berdasarkan hasil pengujian sistem *open loop*, diperoleh bahwa respon level air ketika terdapat aliran *feedwater* sebesar $0,1 \text{ m}^3/\text{s}$ menuju *plant*, maka level air akan bertambah menuju nilai tidak berhingga. Kenaikan level tersebut diakibatkan oleh komponen integrator yang terdapat pada fungsi alih *steam drum*. Jika level air tersebut tidak diatur, maka air di dalam *steam drum* tidak terjaga level 0 m saat beroperasi. Level air diluar batas toleransi dapat membahayakan operasi pada suatu pembangkit, seperti menimbulkan kerusakan pipa *boiler*, sudu turbin, dan

peralatan lainnya. Oleh karena itu, level air *steam drum* membutuhkan adanya kontroler yang mampu menjaga level air berada pada setengah dari ketinggian drum.

4.3 Pengujian Sistem secara *Closed Loop*

Simulasi sistem *open loop* pada sistem pengaturan level air *steam drum* telah dilakukan. Berdasarkan hasil simulasi, sistem *open loop* tidak dapat digunakan karena memberikan nilai level air yang terus meningkat, sehingga dapat membahayakan proses pada pembangkit. Penambahan umpan balik, dapat meningkatkan kestabilan sistem. Pada penelitian ini dilakukan pengujian sistem dengan menambahkan umpan balik level air aktual hasil pengukuran *level transmitter* untuk dibandingkan dengan nilai *setpoint*. Perbedaan nilai pada level aktual dan *setpoint* menghasilkan sinyal kesalahan yang akan terus diperbaiki hingga beda kedua nilai mendekati nol. Diagram blok pengujian sistem *closed loop* dapat dilihat pada Gambar 4.5.



Gambar 4.6. Diagram Blok Pengujian Sistem *Closed Loop*

$G_c(s)$ merupakan fungsi alih kontroler, $G_a(s)$ adalah fungsi alih aktuator yang memiliki persamaan $G_a(s) = K_V \cdot K_L \cdot G_P(s)$, sedangkan $G_D(s)$ yaitu fungsi alih *steam drum*. Berdasarkan Persamaan (3.1), (3.2), (3.3), (3.4), dan (3.5) diperoleh fungsi alih *closed loop* pada sistem pengaturan level air seperti pada Persamaan (4.2).

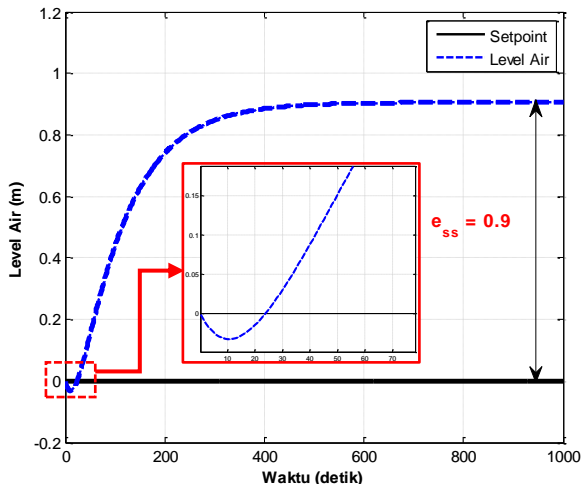
$$G_D(s) = \frac{-0,33s + 0,022}{341,85s^2 + 20,192s + 0,173} \quad (4.2)$$

Fungsi alih *closed loop* pada Persamaan (4.2), memiliki masukan berupa besaran tegangan. Masukan pada sistem *closed loop* tersebut dapat

diubah menjadi tegangan dengan memberikan faktor pengali berupa gain sebesar $7,874 \text{ V/m}$, sehingga diperoleh fungsi alih dengan masukan berupa level seperti pada Persamaan (4.3).

$$G_D(s) = \frac{-2,598s + 0,1732}{341,85s^2 + 20,192s + 0,173} \quad (4.3)$$

Pengujian sistem *closed loop* dilakukan dengan menambahkan gangguan $D(s)$ berupa beban minimal yaitu laju aliran uap yang keluar dari *steam drum*. Laju aliran uap direpresentasikan oleh sinyal *unit step* dengan nilai awal 0 dan nilai akhir 2,21. Respon sistem dengan gangguan pada saat beban minimal dapat dilihat pada Gambar 4.6.

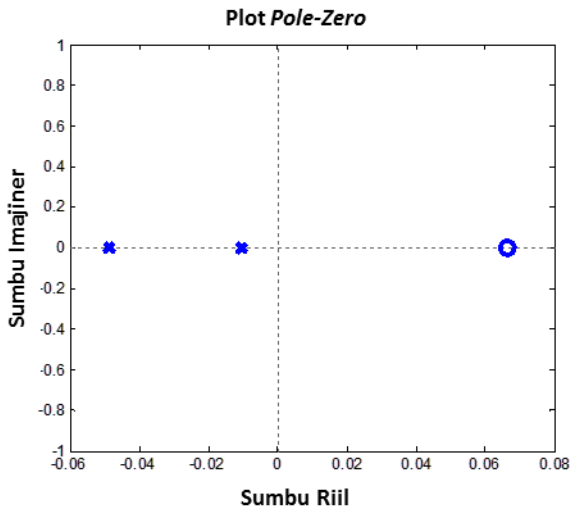


Gambar 4.7. Respon Sistem *Closed Loop* dengan $K_p = 1$, $R(s) = 0$, dan $D(s) = 2,12$

Respon level air *steam drum* pada awalnya menuju arah yang berlawanan dari kondisi tunaknya yang umum disebut sebagai sistem dengan *inverse response*. Sistem yang memiliki letak *zero* di sebelah kanan sumbu imajiner, maka akan memiliki karakteristik tersebut. Sistem pengaturan level air *steam drum* memiliki dua buah *pole* dan *zero* yang terletak pada sumbu riil. Masing-masing *pole* terletak disebelah kiri

sumbu imajiner, yaitu pada -0,0487 dan -0,0104, sedangkan *zero* terletak disebelah kanan sumbu imajiner, yaitu pada 0,0667 yang dapat dilihat pada Gambar 4.7.

Inverse response pada level air *steam drum* umumnya dikenal sebagai fenomena *shrink and swell* (menyusut dan mengembang). Fenomena tersebut terjadi akibat perubahan tekanan di dalam drum mempengaruhi massa jenis air. Air di dalam *steam drum* mengandung gelembung udara layaknya air mendidih. Ketika beban naik, laju aliran air ke dalam *steam drum* akan bertambah. Awalnya level air akan turun karena gelembungnya berkurang, namun setelah itu akan naik kembali.



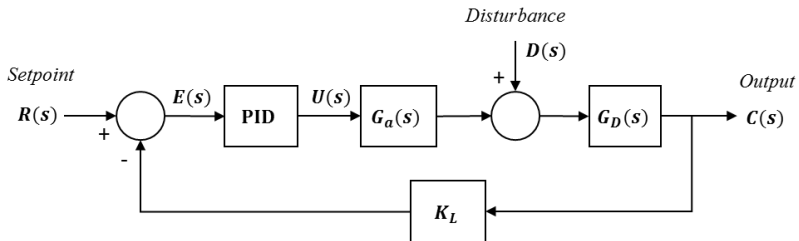
Gambar 4.8. Letak *Pole* dan *Zero* Sistem

Respon sistem dengan penambahan gangguan seperti pada Gambar 4.6 diperoleh kesalahan waktu tunak sebesar 0,9 m; dan respon tidak dapat menuju *setpoint*. Level air melebihi batas normal dapat mengakibatkan pembangkit *trip*. Oleh karena itu, sistem pengaturan level air *steam drum* membutuhkan peran kontroler untuk mengatur level air tetap pada nilai *setpoint* ketika terjadi gangguan.

4.4 Pengujian Sistem dengan Kontroler PID

Penambahan umpan balik pada sistem pengaturan merupakan hal yang sederhana, namun memberikan hasil yang luar biasa. Umpan balik dapat mengurangi efek gangguan dan sensitivitas terhadap perubahan proses. Selain itu, aplikasi umpan balik pada sistem dapat membuat sistem mengikuti *setpoint* yang diberikan. Kontroler PID merupakan salah satu implementasi umpan balik sederhana yang mampu menghilangkan kesalahan kondisi tunak melalui aksi integralnya. Namun, perlu dilakukan pengujian terhadap sistem dengan kontroler PID untuk mengetahui performa sistemnya.

Parameter kontroler PID yang digunakan pada pengujian ini merupakan hasil aturan *tuning* Ziegler-Nichols, yaitu memiliki nilai K_p , τ_i , dan τ_d sebesar 5,26; 47; dan 11,75 berturut-turut. *Setpoint* diberikan nilai nol untuk menyatakan level air yang diinginkan yaitu berada pada setengah dari ketinggian drum (NWL = 0 m). Pengujian dilakukan dengan memberi masukan berupa gangguan yang direpresentasikan oleh sinyal uji *unit step* dengan nilai akhir sesuai Tabel 4.1.



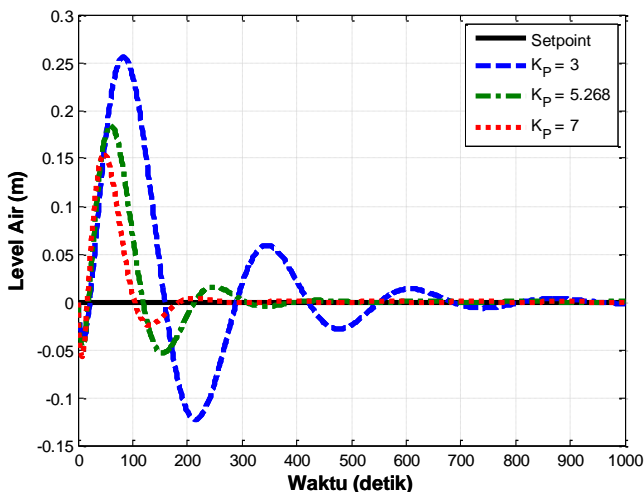
Gambar 4.9. Diagram Blok Sistem Pengaturan dengan Kontroler PID

Pada penelitian ini dilakukan pengujian pada sistem pengaturan level air *steam drum* dengan menggunakan kontroler PID yang telah dirancang pada Bab 3. Pengujian dilakukan dengan melakukan beberapa simulasi antara lain:

- 1) Pengujian dengan variasi nilai K_p
- 2) Pengujian dengan variasi nilai τ_i
- 3) Pengujian dengan variasi nilai τ_d
- 4) Pengujian dengan beban tetap
- 5) Pengujian dengan beban acak
- 6) Pengujian dengan perubahan kriteria beban

4.4.1 Pengujian dengan Variasi Nilai K_p

Pengujian sistem dengan kontroler PID terhadap variasi nilai gain proporsional, K_p , bertujuan untuk memahami dan menganalisis respon sistem ketika gain proporsionalnya berubah. Pengujian dilakukan dengan memberikan *setpoint* level air *steam drum* pada titik NWL, $R(s) = 0$. Kemudian diberikan beban minimal berupa laju aliran uap keluar dari drum, $D(s) = 2,21$. Parameter K_p memberikan aksi perbaikan yang sebanding dengan kesalahan pada sistem. Jika nilai K_p semakin besar, maka aksi perbaikan kontroler terhadap kesalahan semakin cepat. Jika sistem hanya menggunakan kontroler proporsional, maka penambahan nilai K_p dapat mengurangi kesalahan keadaan tunak serta mempercepat respon sistem.



Gambar 4.10. Respon Sistem Menggunakan Kontroler PID dengan Variasi K_p

Pada umumnya, seseorang menginginkan aksi perbaikan kontroler terhadap kesalahan terjadi dalam waktu yang singkat. Berdasarkan keinginan tersebut, aksi perbaikan kesalahan pada sistem dapat dipercepat dengan meningkatkan nilai gain proporsional. Namun, perubahan nilai K_p tidak hanya mempengaruhi durasi perbaikan kesalahan, akan tetapi juga menyebabkan respon sistem berosilasi. Hal itu terjadi akibat aksi

perbaikan terhadap kesalahan terlalu cepat sehingga respon sistem akan osilasi terlebih dahulu sebelum sampai pada kondisi tunaknya.

Tabel 4.2. Spesifikasi Respon Menggunakan Kontroler PID dengan Variasi K_p

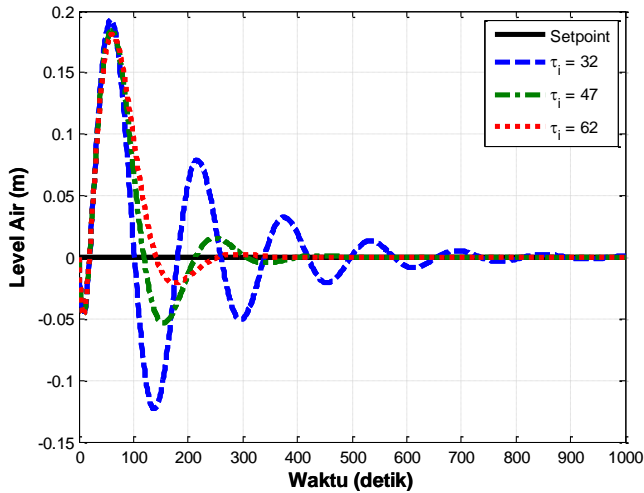
Variasi K_p	ISE	IAE	ITAE	ρ (m)	t_{rs} (detik)
$K_p = 3$	345,2	326,3	$6,6 \times 10^4$	0,26	920
$K_p = 5,268$	108	126,5	$1,3 \times 10^4$	0,19	420
$K_p = 7$	59,5	76,9	5036	0,15	260

Pengujian dengan variasi nilai K_p dilakukan melalui simulasi pada sistem pengaturan level air *steam drum* dengan kontroler PID, sedangkan nilai τ_i , dan τ_d dibuat konstan, yaitu sebesar 47 dan 11,75. Hasil pengujian sistem dengan variasi nilai K_p dapat dilihat pada Gambar 4.5. Berdasarkan hasil simulasi, diperoleh kesimpulan bahwa nilai parameter K_p dapat mempercepat respon sistem, mengurangi kesalahan keadaan tunak, dan dapat menimbulkan osilasi pada sistem. Perbandingan indeks performansi kesalahan, *perturbation peak*, dan *recovery time* ketika nilai K_p divariasikan dapat dilihat pada Tabel 4.2.

4.4.2 Pengujian dengan Variasi Nilai τ_i

Pengujian sistem dengan kontroler PID terhadap variasi nilai waktu integral, τ_i bertujuan untuk mengetahui dan menganalisis respon sistem ketika parameter waktu integral mengalami perubahan. Fungsi utama bagian integral pada kontroler tersebut adalah untuk menghilangkan kesalahan pada keadaan tunak. Hilangnya kesalahan pada keadaan tunak dapat terjadi, karena bagian integral akan memberikan aksi perbaikan kesalahan walaupun kesalahan yang terjadi sangat kecil. Aksi perbaikan kontroler terhadap kesalahan akan konstan, apabila kesalahan pada sistem sama dengan nol.

Kontroler membutuhkan waktu perbaikan yang berbeda-beda untuk menghilangkan kesalahan pada keadaan tunak. Hal ini dipengaruhi oleh besarnya nilai parameter waktu integral pada kontroler. Pengujian dengan variasi nilai τ_i dilakukan melalui simulasi pada sistem pengaturan level air *steam drum* menggunakan kontroler PID dengan nilai K_p dan τ_d sebesar 5,26 dan 11,75. Hasil simulasi pengujian sistem dengan variasi nilai waktu integral dapat dilihat pada Gambar 4.9.



Gambar 4.11. Respon Sistem Menggunakan Kontroler PID dengan Variasi τ_i

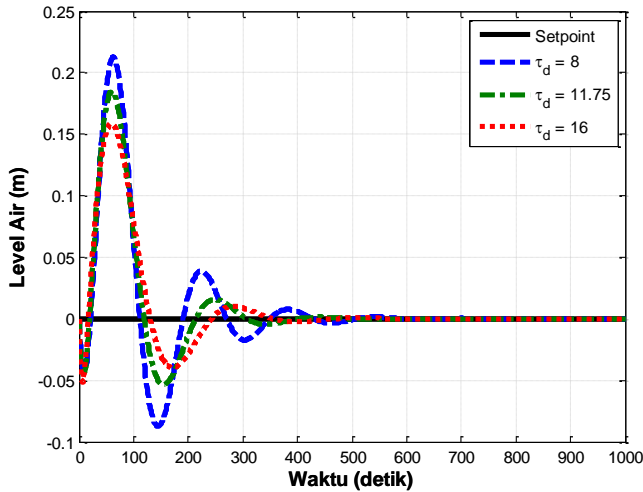
Berdasarkan hasil simulasi, diperoleh kesimpulan apabila nilai τ_i semakin besar, maka respon sistem semakin lambat dan osilasi sistem berkurang. Perbandingan indeks performansi kesalahan, *perturbation peak*, dan *recovery time* ketika nilai τ_i divariasikan dapat dilihat pada Tabel 4.3.

Tabel 4.3. Spesifikasi Respon Menggunakan Kontroler PID dengan Variasi τ_i

Variasi τ_i	ISE	IAE	ITAE	ρ (m)	t_{rs} (detik)
$\tau_i = 32$	156,6	217,4	$4,2 \times 10^4$	0,193	300
$\tau_i = 47$	108,1	126,5	$1,3 \times 10^4$	0,185	438
$\tau_i = 62$	108,8	115	9400	0,181	845

4.4.3 Pengujian dengan Variasi Nilai τ_d

Pengujian sistem dengan kontroler PID terhadap variasi nilai parameter waktu derivatif dilakukan untuk memahami dan menganalisis respon sistem apabila parameter waktu derivatif mengalami perubahan. Aksi derivatif pada kontroler tersebut bertujuan untuk meningkatkan kestabilan sistem *closed loop*. Aksi kontrol membutuhkan waktu untuk memberikan efek perubahan pada sistem, hal ini terjadi karena adanya dinamika proses pada sistem tersebut. Keterlambatan aksi kontrol dalam memperbaiki kesalahan dapat menyebabkan sistem tidak stabil.



Gambar 4.12. Respon Sistem Menggunakan Kontroler PID dengan Variasi τ_d

Pengujian dengan variasi nilai waktu derivatif dilakukan melalui simulasi pada sistem pengaturan level air *steam drum* dengan kontroler PID. Hasil pengujian sistem dengan variasi nilai waktu derivatif dapat dilihat pada Gambar 4.8. Apabila parameter waktu derivatif semakin besar, maka osilasi pada sistem semakin berkurang. Hal ini disebabkan oleh aksi dari komponen derivatif, yaitu mengurangi osilasi pada sistem. Perbandingan indeks performansi kesalahan, *perturbation peak*, dan *recovery time* ketika nilai τ_d divariasikan dapat dilihat pada Tabel 4.3.

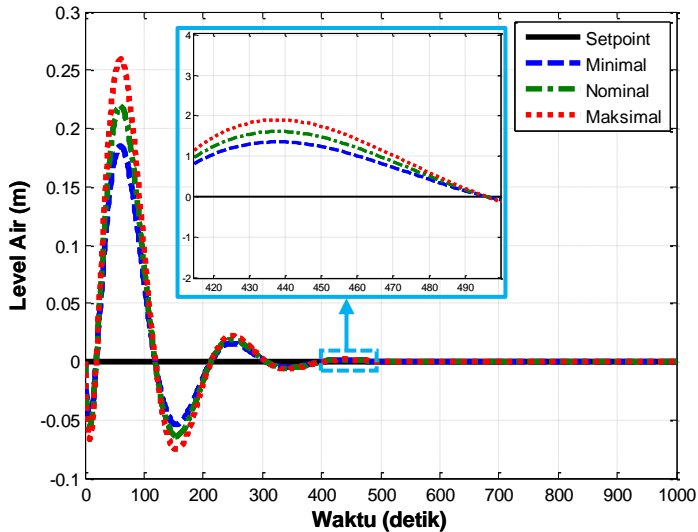
Tabel 4.4. Spesifikasi Respon Menggunakan Kontroler PID dengan Variasi τ_d

Variasi τ_d	ISE	IAE	ITAE	ρ (m)	t_{rs} (detik)
$\tau_d = 8$	143,7	158,5	$1,9 \times 10^4$	0,21	500
$\tau_d = 11.75$	107,9	126,4	$1,26 \times 10^4$	0,18	415
$\tau_d = 16$	89,3	118,3	$1,9 \times 10^4$	0,16	365

4.4.4 Pengujian dengan Beban Tetap

Pengujian sistem dengan kontroler PID terhadap beban tetap bertujuan untuk menganalisis respon sistem apabila menerima perubahan

beban secara tiba-tiba, kemudian bertahan pada kondisi tersebut sampai waktu tertentu. Hal ini merepresentasikan kondisi riil, ketika pihak P3B meminta daya pembangkit lebih tinggi, maka laju aliran uap keluar dari drum akan meningkat. Hal ini mengakibatkan level air *steam drum* menurun. Sistem pengaturan level air *steam drum* dengan kontroler PID mampu membawa level air kembali menuju *setpoint* ketika diberi gangguan. Penggunaan kontroler PID mampu mengeliminasi kesalahan kondisi tunak, karena memiliki komponen integral. Kesalahan akan terus diperbaiki oleh komponen tersebut, walaupun nilai kesalahannya kecil. Besarnya beban pembangkit bervariasi, sehingga dilakukan pengujian dengan beban berbeda, yaitu beban minimal, nominal, dan maksimal. Masing-masing beban tersebut berupa laju aliran uap keluar dari drum sebesar 2,21; 2,63; dan 3,10 m³/s.



Gambar 4.13. Respon Sistem Menggunakan Kontroler PID dengan Beban Tetap

Pengujian dilakukan dengan memberi gangguan berupa laju aliran uap keluar dari drum yang direpresentasikan oleh sinyal *step*, sehingga diperoleh respon sistem seperti pada Gambar 4.11. Ketika sistem diberikan beban minimal, nominal, dan maksimal, maka diperoleh respon sistem memiliki *perturbation peak* sebesar 0,18 m; 0,22 m; dan 0,26 m

berturut. Waktu yang dibutuhkan respon sistem dengan kontroler PID untuk menuju nilai *setpoint* ketika menerima beban dapat diamati melalui besarnya *recovery time* seperti pada Tabel 4.5. Semakin besar beban yang diberikan, maka semakin lama waktu yang dibutuhkan sistem untuk menuju nilai *setpoint*. Indeks performansi kesalahan semakin besar, ketika beban yang diterima oleh sistem semakin besar.

Tabel 4.5. Spesifikasi Respon Menggunakan Kontroler PID dengan Beban Tetap

Beban	ISE	IAE	ITAE	ρ (m)	t_{rs} (detik)
Minimal	108,1	126,5	$1,3 \times 10^4$	0,18	330
Nominal	153,1	150,6	$1,5 \times 10^4$	0,22	324
Maksimal	212,7	177,5	$1,8 \times 10^4$	0,26	320

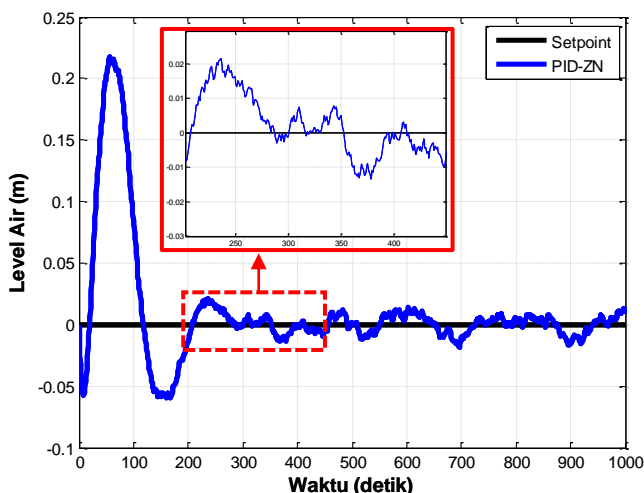
4.4.5 Pengujian dengan Beban Acak

Pengujian sistem dengan kontroler PID terhadap beban acak bertujuan untuk memahami dan menganalisis respon sistem ketika menerima gangguan berupa efek beban acak. Sistem pengaturan level air *steam drum* mengalami efek beban ketika terdapat laju aliran uap keluar dari drum. Laju aliran uap yang keluar dari *steam drum* umumnya tidak konstan, melainkan berubah-ubah pada rentang nilai tertentu karena kondisi riil sistem tidak ideal. Pengujian dilakukan dengan memberi gangguan berupa efek beban acak yang direpresentasikan oleh *uniform noise* dengan nilai acak antara 2,21 sampai 3,1 untuk mendeskripsikan respon level *steam drum* ketika beban bervariasi pada nilai kriteria beban.

Respon level air *steam drum* menggunakan kontroler PID mampu membawa level air kembali menuju *setpoint*, namun level air osilasi disekitar *setpoint* seperti pada Gambar 4.12. Berdasarkan hasil simulasi ketika diberi gangguan berupa efek beban laju aliran uap minimal, nominal, dan maksimal memiliki *perturbation peak* sebesar 0,22 m dan *recovery time* sebesar 296 detik. Indeks performansi kesalahan sistem ketika menerima beban acak, yaitu sebesar 154,1; 176,8; dan $3,1 \times 10^4$ untuk ISE, IAE, dan ITAE berturut-turut.

Tabel 4.6. Spesifikasi Respon Menggunakan Kontroler PID dengan Beban Acak

Beban	ISE	IAE	ITAE	ρ (m)	t_{rs} (detik)
Acak	154,1	176,8	$3,1 \times 10^4$	0,22	296



Gambar 4.14. Respon Sistem Menggunakan Kontroler PID dengan Beban Acak

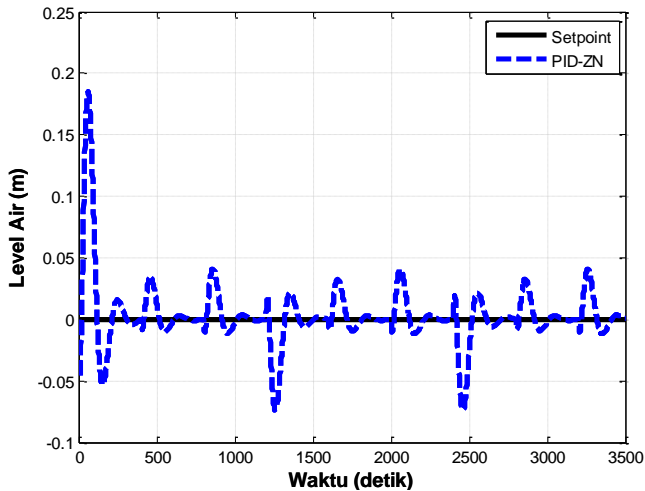
4.4.6 Pengujian dengan Perubahan Kriteria Beban

Pengujian ini dilakukan pada sistem pengaturan level air *steam drum* menggunakan kontroler PID bertujuan untuk menganalisis performa sistem apabila menerima beban yang berubah nilai akhirnya. Pada kondisi riil, hal ini dapat menggambarkan variasi beban pembangkit selama satu hari. Pengujian dengan dilakukan dengan memberi nilai *setpoint* sama dengan nol dan memberi nilai gangguan berupa aliran uap keluar dari drum yang berubah-ubah seperti pada Gambar 4.3.

Sistem dengan kontroler PID ketika menerima gangguan memberikan respon level air seperti pada Gambar 4.15. Berdasarkan hasil simulasi diperoleh kesimpulan bahwa sistem pengaturan tersebut mampu meredam gangguan berupa beban berubah. Selain itu, diperoleh bahwa indeks performansi kesalahan dari sistem pengaturan level air *steam drum* menggunakan kontroler PID ketika menerima beban berubah adalah sebesar 169,7; 378,4; dan $4,9 \times 10^5$ untuk ISE, IAE, dan ITAE berurutan.

Tabel 4.7. Spesifikasi Respon Menggunakan Kontroler PID dengan Perubahan Kriteria Beban

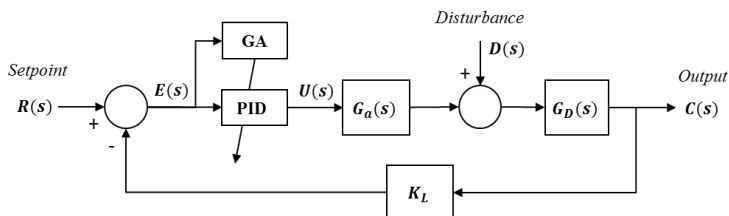
Beban	ISE	IAE	ITAE	ρ (m)	t_{rs} (detik)
Berubah	169,7	378,4	$4,9 \times 10^5$	0,185	340



Gambar 4.15. Respon Sistem Menggunakan Kontroler PID dengan Perubahan Kriteria Beban

4.5 Pengujian Sistem dengan Kontroler PID-GA

Kontroler PID-GA merupakan modifikasi dari kontroler PID konvensional dengan menambahkan mekanisme *tuning* parameter menggunakan GA. Diagram blok pengujian sistem dengan kontroler PID-GA dapat dilihat pada Gambar 4.16 Parameter yang diperoleh dari mekanisme tersebut diharapkan merupakan parameter yang optimal berdasarkan indeks performansi yang digunakan adalah kombinasi dari ISE, IAE, dan ITAE.



Gambar 4.16. Diagram Blok Sistem Pengaturan Menggunakan Kontrol PID-GA

Pengujian sistem dengan kontroler PID-GA dilakukan untuk menganalisis performa sistem setelah ditambahkan mekanisme *tuning*

parameter pada kontroler PID. *Setpoint* pada pengujian ini diatur pada nilai nol untuk merepresentasikan level air *steam drum* berada pada setengah dari ketinggian drum. Sistem diberi masukan berupa gangguan dengan nilai sesuai pada Tabel 4.1. Parameter PID dikodekan dalam bilangan biner dengan masing-masing memiliki ukuran 10 bit. Probabilitas *crossover* dan mutasi yang digunakan sebesar 0,6 dan 0,02 seperti pada Tabel 4.8.

Tabel 4.8. Parameter yang Digunakan pada GA

Parameter	Tipe/Nilai
Tipe GA	<i>Simple GA</i>
Tipe Seleksi	<i>Tournament</i>
Tipe <i>Crossover</i>	<i>Single point</i>
Tipe Mutasi	<i>Flip</i>
Generasi maksimum	40 generasi
Ukuran populasi	30 individu
Ukuran kromosom	30 bit
Probabilitas <i>crossover</i>	0,6
Probabilitas mutasi	0,02

Pada penelitian ini dilakukan pengujian pada sistem pengaturan level air *steam drum* dengan menggunakan kontroler PID-GA yang telah dirancang pada Bab 3. Pengujian dilakukan dengan melakukan beberapa simulasi antara lain:

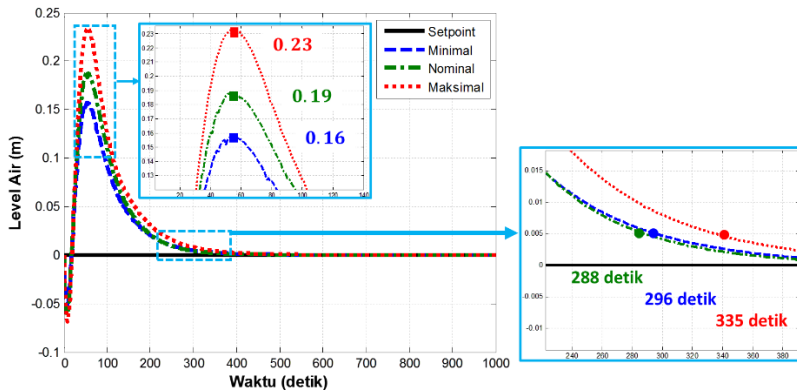
- 1) Pengujian dengan beban tetap
- 2) Pengujian dengan beban acak
- 3) Pengujian dengan perubahan kriteria beban
- 4) Pengujian dengan variasi indeks performansi
- 5) Pengujian dengan variasi parameter GA

4.5.1 Pengujian dengan Beban Tetap

Sistem pengaturan level air *steam drum* memiliki gangguan berupa efek beban yang diberikan oleh laju aliran uap keluar dari drum. Gangguan tersebut dapat berubah secara tiba-tiba ketika *dispatcher* atau P3B melakukan perubahan permintaan daya pada pembangkit. Kontroler PID-GA yang telah dirancang pada Bab 3 diterapkan pada sistem tersebut, kemudian dilakukan pengujian dengan memberikan beban tetap. Kriteria beban yang digunakan pada penelitian ini sesuai dengan Tabel 4.1, yaitu

beban minimal, nominal, dan maksimal dengan laju aliran uap keluar dari drum sebesar 2,21; 2.63; dan 3,10 m³/s.

Perubahan beban tetap direpresentasikan oleh sinyal *unit step* seperti pada Gambar 4.1. Respon level air *steam drum* menggunakan kontroler PID-GA memiliki *perturbation peak* yang lebih kecil dibandingkan dengan kontroler PID konvensional. Sistem dengan menggunakan kontroler PID-GA secara urut memiliki *perturbation peak* sebesar 0,16 m; 0,19 m; dan 0,23 m ketika menerima beban minimal, nominal, dan maksimal. Selain itu, sistem dengan kontroler PID-GA juga memiliki *recovery time* yang lebih cepat daripada sistem dengan kontroler PID konvensional. Berdasarkan hasil simulasi sistem dengan kontroler PID-GA, respon sistem secara berurutan memiliki *recovery time* sebesar 296 detik; 288 detik; dan 335 detik ketika menerima beban minimal, nominal, dan maksimal. Respon level air *steam drum* seperti pada Gambar 4.17.



Gambar 4.17. Respon Menggunakan Kontroler PID-GA dengan Beban Tetap

Sistem pengaturan level air *steam drum* menggunakan kontroler PID-GA memberikan nilai indeks performansi kesalahan yang lebih kecil daripada sistem dengan kontroler PID konvensional. Hasil simulasi pada penelitian ini memberikan nilai indeks performansi kesalahan seperti pada Tabel 4.9, sehingga kontroler PID-GA memiliki kesalahan yang lebih kecil daripada kontroler PID konvensional.

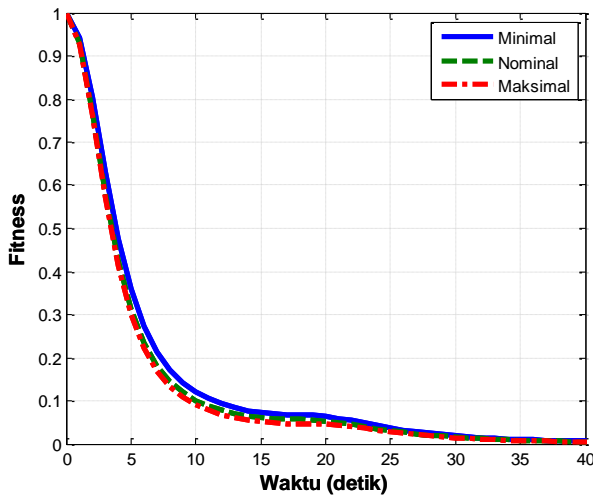
Fungsi objektif kontroler PID-GA pada penelitian ini adalah meminimalkan kombinasi dari ISE, IAE, dan ITAE. Fungsi *fitness* sesuai

Persamaan (2.30) bertujuan untuk memperoleh nilai maksimal yang merepresentasikan nilai ketahanan individu terbaik pada GA.

Tabel 4.9. Spesifikasi Respon Sistem Menggunakan Kontroler PID-GA dengan Beban Tetap

Beban	ISE	IAE	ITAE	ρ (m)	t_{rs} (detik)
Minimal	96,9	129,2	$1,3 \times 10^4$	0,16	296
Nominal	136,7	149,7	$1,4 \times 10^4$	0,19	288
Maksimal	200,4	186,2	$1,9 \times 10^4$	0,23	335

Peran GA pada kontroler tersebut adalah mencari nilai parameter K_p , τ_i , dan τ_d yang meminimalkan indeks performansi kesalahan atau dengan kata lain memaksimalkan fungsi *fitness*. Berdasarkan hasil simulasi, beban maksimal memiliki konvergensi nilai *fitness* yang lebih cepat dibandingkan beban minimal dan nominal seperti pada Gambar 4.18.

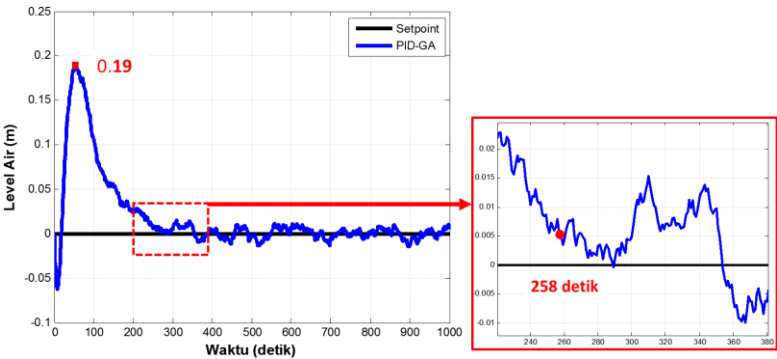


Gambar 4.18. Nilai *Fitness* pada Kontroler PID-GA dengan Beban Tetap

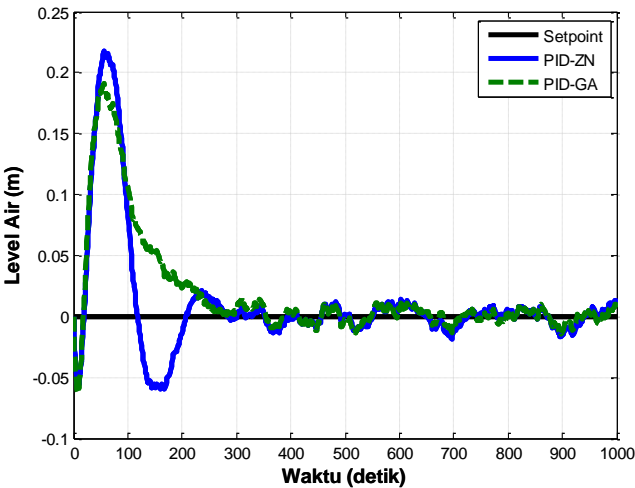
4.5.2 Pengujian dengan Beban Acak

Pengujian dengan beban acak pada sistem menggunakan kontroler PID-GA bertujuan untuk menganalisis performa sistem menggunakan kontroler tersebut. Penambahan mekanisme *tuning* parameter dengan

fungsi objektif meminimalkan indeks performansi kesalahan memiliki performa sistem lebih baik dibandingkan dengan kontroler PID konvensional. Pada penelitian ini dilakukan simulasi pengujian sistem pengaturan level air *steam drum* menggunakan kontroler PID-GA.



Gambar 4.19. Respon Menggunakan Kontroler PID-GA dengan Beban Acak



Gambar 4.20. Respon Menggunakan Kontroler PID dan PID-GA dengan Beban Acak

Hasil simulasi sistem menggunakan kontroler PID-GA dengan beban acak memperoleh respon seperti pada Gambar 4.19 dengan *perturbation peak* sebesar 0,19 m dan *recovery time* sebesar 258 detik. Perbandingan respon sistem dengan kontroler PID dan PID-GA dapat dilihat pada Gambar 4.20. Dengan demikian respon sistem dengan kontroler PID-GA memiliki kemampuan meredam gangguan yang lebih baik daripada kontroler PID konvensional yang ditunjukkan dengan nilai *perturbation peak* yang lebih kecil dan waktu kembali menuju *setpoint* yang lebih cepat. Indeks performansi kesalahan sistem dengan kontroler PID-GA adalah sebesar 137,3; 176,7; dan $3,1 \times 10^4$ untuk ISE, IAE, dan ITAE seperti pada Tabel 4.10. Indeks performansi kesalahan sistem dengan kontroler PID konvensional dengan beban acak, yaitu sebesar 154,1; 176,8; dan $3,1 \times 10^4$ untuk ISE, IAE, dan ITAE berturut-turut.

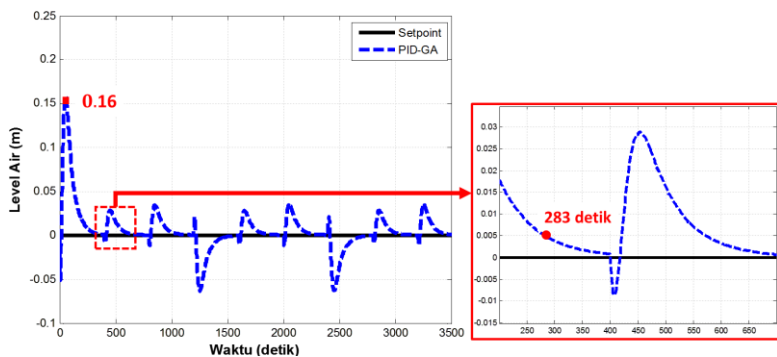
Tabel 4.10. Spesifikasi Respon Sistem Menggunakan Kontroler PID-GA dan PID dengan Beban Acak

Kontroler	ISE	IAE	ITAE	ρ (m)	t_{rs} (detik)
PID-GA	137,5	174,4	$3,1 \times 10^4$	0.19	250
PID-ZN	154,1	176,8	$3,1 \times 10^4$	0.22	296

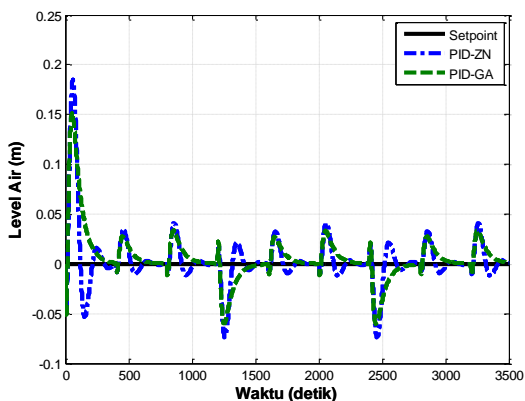
4.5.3 Pengujian dengan Perubahan Kriteria Beban

Pengujian dengan perubahan kriteria beban dilakukan untuk menganalisis performa sistem dengan menggunakan kontroler PID-GA apabila menerima gangguan berupa beban berubah. Level air *steam drum* mampu kembali menuju nilai *setpoint* ketika diberi gangguan berupa beban berubah. Beban berubah dengan pola beban minimal, nominal, dan maksimal secara berurutan. Hal ini dilakukan untuk merepresentasikan beban pada riil sistem yang berubah-ubah selama pembangkit beroperasi.

Berdasarkan hasil indeks perfomansi kesalahan sistem dengan kontroler PID-GA adalah sebesar 147,2; 373; dan $4,9 \times 10^5$ untuk ISE, IAE, dan ITAE dapat dilihat pada Tabel 4.11. Sistem tersebut memiliki *perturbation peak* sebesar 0,16 m dan *recovery time* sebesar 283 detik dengan respon seperti pada Gambar 4.21. Perbandingan respon kontroler PID dan PID-GA dapat dilihat pada Gambar 4.22. Dengan demikian, sistem pengaturan level air *steam drum* menggunakan kontroler PID-GA memiliki respon sistem lebih baik daripada kontroler PID konvensional. Indeks performansi kesalahan sistem dengan kontroler PID konvensional yaitu sebesar 169,7; 378,4; dan $4,9 \times 10^5$ untuk ISE, IAE, dan ITAE berurutan.



Gambar 4.21. Respon Menggunakan Kontroler PID-GA dengan Perubahan Kriteria Beban



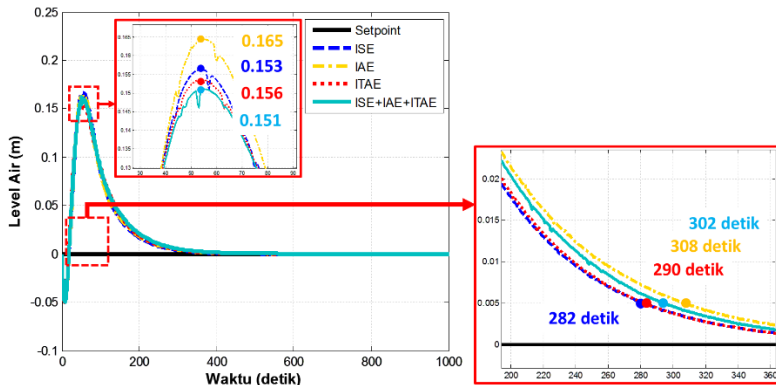
Gambar 4.22. Respon Menggunakan Kontroler PID dan PID-GA dengan Perubahan Kriteria Beban

Tabel 4.11. Spesifikasi Respon Sistem Menggunakan Kontroler PID-GA dan PID dengan Perubahan Kriteria Beban

Kontroler	ISE	IAE	ITAE	ρ (m)	t_{rs} (detik)
PID-GA	147,2	373	$4,9 \times 10^5$	0,16	283
PID	169,7	378,4	$4,9 \times 10^5$	0,185	340

4.5.4 Pengujian dengan Variasi Fungsi Objektif

Kontroler PID-GA memiliki fungsi objektif yang ditentukan ketika mendesain kontroler tersebut. Pengujian dengan variasi fungsi objektif pada kontroler PID-GA dilakukan untuk menganalisis performa sistem terhadap perubahan fungsi objektif. Fungsi objektif yang digunakan pada penelitian ini adalah meminimalkan indeks performansi kesalahan, yaitu ISE, IAE, ITAE, dan kombinasi ketiganya. Parameter kontroler PID dari GA adalah nilai optimal yang meminimalkan fungsi objektifnya. Parameter kontroler PID-GA yang digunakan pada pengujian ini sesuai pada Tabel 4.6, namun pada bagian fungsi objektifnya dilakukan variasi. *Setpoint* diatur pada nilai nol, kemudian diberi masukan berupa beban nominal, yaitu laju aliran uap sebesar $2,63 \text{ m}^3/\text{s}$, sehingga diperoleh respon seperti pada Gambar 4.23.



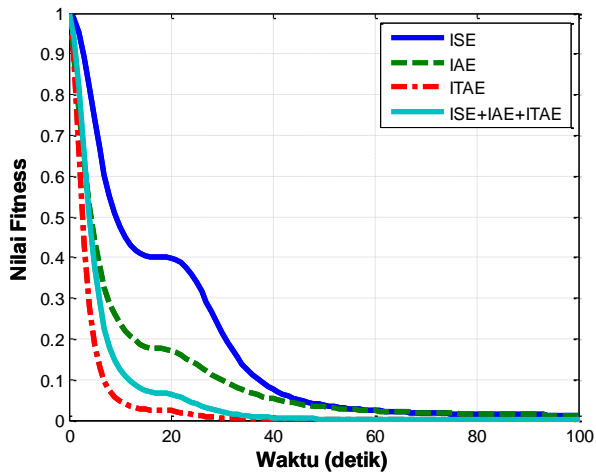
Gambar 4.23. Respon Level Air *Steam Drum* Menggunakan Kontroler PID-GA dengan Variasi Fungsi Objektif

Fungsi objektif yang digunakan pada kontroler PID-GA memiliki pengaruh pada besarnya nilai *perturbation peak* dan *recovery time*. Hal ini disebabkan oleh nilai parameter kontroler PID yang diperoleh dari mekanisme *tuning* menggunakan GA meminimalkan indeks performansi yang berbeda-beda. Berdasarkan Tabel 4.12, spesifikasi respon sistem dengan fungsi objektif kombinasi dari ISE, IAE, dan ITAE memiliki *perturbation peak* paling kecil, sedangkan fungsi objektif ITAE memiliki *recovery time* paling cepat. *Recovery time* sistem dengan fungsi objektif

ITAE yang cepat, didukung oleh nilai *fitness*-nya yang lebih cepat menuju nol seperti pada Gambar 4.24.

Tabel 4.12. Spesifikasi Respon Sistem Menggunakan Kontroler PID-GA dengan Variasi Fungsi Objektif

Fungsi Objektif	ISE	IAE	ITAE	ρ (m)	t_{rs} (detik)
ISE	99,2	122,9	$1,2 \times 10^4$	0,153	282
IAE	99,4	130,7	$1,3 \times 10^4$	0,165	308
ITAE	92,9	124,5	$1,2 \times 10^4$	0,156	290
ISE+IAE+ITAE	93,2	127,6	$1,2 \times 10^4$	0,151	302

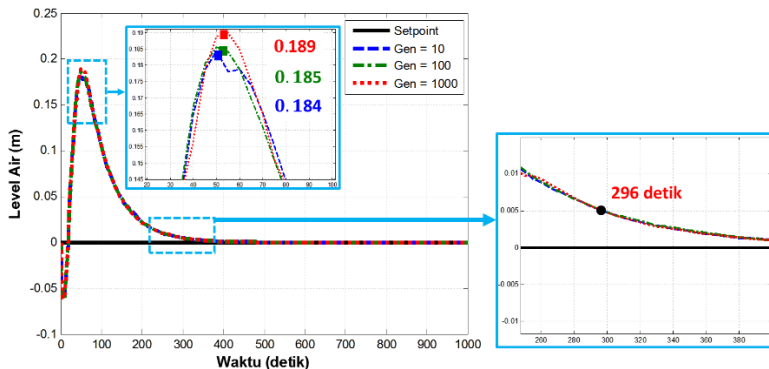


Gambar 4.24. Nilai *Fitness* pada Kontroler PID-GA dengan Variasi Fungsi Objektif

4.5.5 Pengujian dengan Variasi Jumlah Generasi

Pengujian ini dilakukan untuk menganalisis performa sistem menggunakan kontroler PID-GA terhadap perubahan jumlah generasi. Tipe dan besarnya nilai parameter GA selain jumlah generasi mengikuti konfigurasi seperti pada Tabel 4.6. *Setpoint* diatur pada nilai nol, sedangkan masukan sistem adalah beban nominal, yaitu 2,63 m³/s berupa sinyal *unit step* dengan nilai akhirnya sebesar 2,63.

Repon sistem ketika menerima gangguan dari efek beban nominal memiliki *perturbation peak* sebesar 0,189 m; 0,185 m; dan 0,184 m untuk jumlah generasi yang digunakan 10; 100; dan 1000 berturut-turut dapat dilihat pada Gambar 4.25. *Recovery time* ketika menerima beban yang sama, setiap variasi jumlah generasi memiliki nilai yang sama yaitu 296 detik sesuai data pada Tabel 4.13. Hal ini dikarenakan jumlah generasi tidak mempengaruhi perubahan nilai *fitness* setiap calon parameter kontroler yang terpilih. Namun, jumlah generasi tersebut berpengaruh pada banyaknya iterasi pada komputasi GA untuk memperoleh solusi yang optimal terhadap fungsi objektifnya. Berdasarkan hasil simulasi, variasi jumlah generasi tidak memberikan pengaruh signifikan pada spesifikasi respon sistem. Namun, jumlah generasi tersebut berpengaruh pada lamanya waktu komputasi. Lama waktu simulasi yang dibutuhkan untuk 10 generasi adalah 71 detik, 100 generasi adalah 600 detik, dan 1000 generasi adalah 5839 detik.



Gambar 4.25. Respon Level Air *Steam Drum* Menggunakan Kontroler PID-GA dengan Variasi Jumlah Generasi

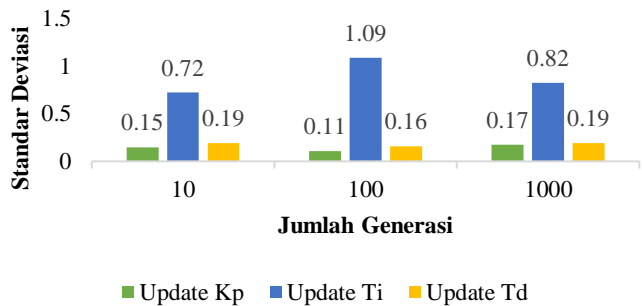
Nilai *update* parameter kontroler pada sistem riil membutuhkan waktu yang cepat untuk memenuhi spesifikasi yang diinginkan. Waktu komputasi terlalu lama, menyebabkan keterlambatan *update* nilai parameter kontroler PID, sehingga parameter yang dihasilkan tidak sesuai lagi dengan kondisi riil saat ini.

Mekanisme *tuning* parameter menggunakan GA memberikan *update* parameter ketika syarat berhenti algoritma terpenuhi. Parameter PID yang dihasilkan merupakan parameter optimal yang meminimalkan

fungsi objektif, atau memiliki nilai *fitness* tertinggi diantara populasi pada saat itu. Jumlah generasi pada GA tidak memiliki pengaruh spesifik pada besarnya perubahan nilai parameter kontroler dapat dilihat pada Gambar 4.26. Dengan demikian, jumlah generasi semakin besar tidak menjamin perubahan parameter semakin sering terjadi.

Tabel 4.13. Spesifikasi Respon Sistem Menggunakan Kontroler PID-GA dengan Variasi Jumlah Generasi

Parameter GA	ISE	IAE	ITAE	ρ (m)	t_{rs} (detik)
Generasi = 10	26,3	29,7	2865	0,184	296
Generasi = 100	26,2	29,6	2859	0,185	296
Generasi = 1000	26,7	29,8	2858	0,189	296



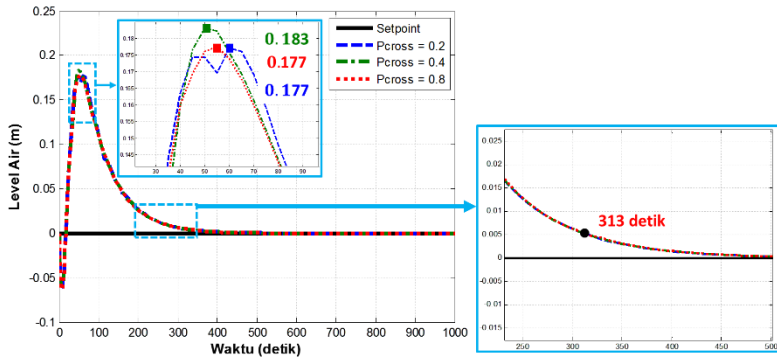
Gambar 4.26. Update Parameter Kontroler PID-GA dengan Variasi Jumlah Generasi

4.5.6 Pengujian dengan Variasi Probabilitas *Crossover*

Operator genetika yang digunakan pada GA salah satunya adalah *crossover*, sehingga pada penelitian ini dilakukan pengujian pada parameter tersebut. Pengujian ini bertujuan untuk menganalisis respon sistem ketika parameter probabilitas *crossover* yang digunakan pada kontroler PID-GA memiliki nilai berbeda-beda. Tipe dan nilai parameter GA selain probabilitas *crossover* diatur sesuai pada Tabel 4.6. Respon pada Gambar 4.27 diperoleh ketika sistem memiliki nilai *setpoint* sebesar nol, kemudian diberi gangguan berupa efek beban nominal yang menyatakan laju aliran uap sebesar 2,63 m³/s.

Spesifikasi respon level air *steam drum* tidak berubah secara signifikan, ketika probabilitas *crossover* pada kontroler PID-GA

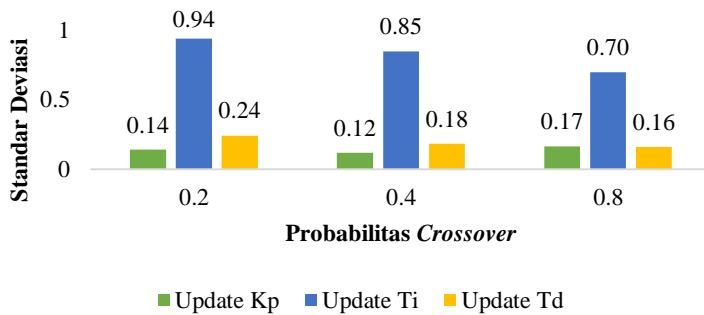
divariasikan. Berdasarkan hasil simulasi diperoleh data spesifikasi respon sistem seperti pada Tabel 4.14, dapat diamati bahwa nilai indeks performansi, *perturbation peak*, dan *recovery time* tidak berbeda jauh. Jika ditinjau dari nilai *fitness* selama simulasi, maka variasi probabilitas *crossover* tidak memiliki perbedaan yang signifikan sesuai pada Gambar 4.28. Standar deviasi parameter terdapat pada Gambar 4.29.



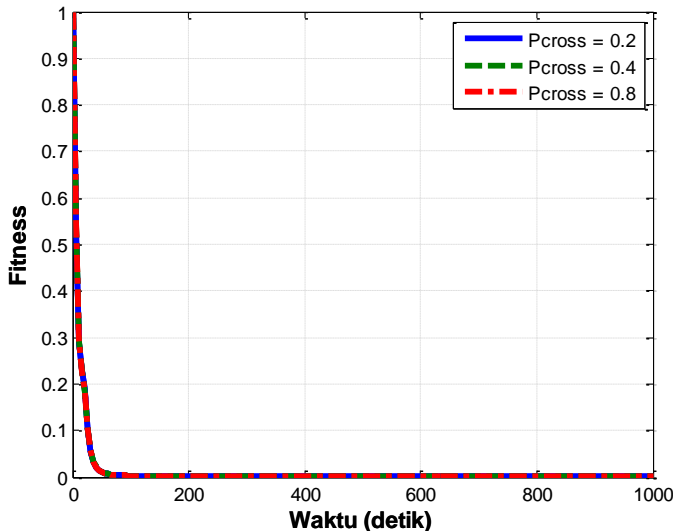
Gambar 4.27. Respon Level Air *Steam Drum* Menggunakan Kontroler PID-GA dengan Variasi Probabilitas *Crossover*

Tabel 4.14. Spesifikasi Respon Sistem Menggunakan Kontroler PID-GA dengan Variasi Probabilitas *Crossover*

Parameter GA	ISE	IAE	ITAE	ρ (m)	t_{rs} (detik)
PCross = 0.2	26,4	30,59	3079	0,179	313
PCross = 0.4	26,3	30,56	3082	0,181	313
PCross = 0.8	26,6	20,66	3081	0,183	313



Gambar 4.28. Update Parameter Kontroler PID-GA dengan Variasi Probabilitas Crossover



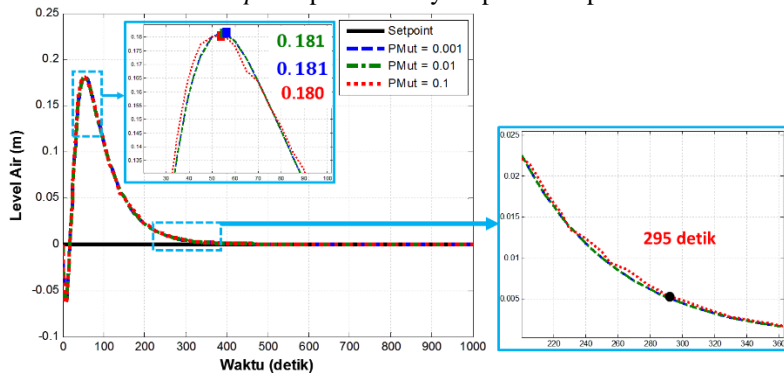
Gambar 4.29. Nilai *Fitness* dengan Variasi Probabilitas Crossover

4.5.7 Pengujian dengan Variasi Probabilitas Mutasi

Probabilitas mutasi merupakan salah satu operator pada GA yang berperan untuk memperoleh solusi optimal terhadap fungsi objektifnya. Pada penelitian ini dilakukan pengujian terhadap variasi probabilitas mutasi yang digunakan pada kontroler PID-GA. Hal tersebut bertujuan untuk menganalisis performa sistem terhadap perubahan probabilitas

mutasi. Ketika *setpoint* diatur pada nilai nol serta diberi beban nominal berupa laju aliran uap sebesar $2,63 \text{ m}^3/\text{s}$, maka respon level air *steam drum* dengan variasi probabilitas mutasi seperti pada Gambar 4.30.

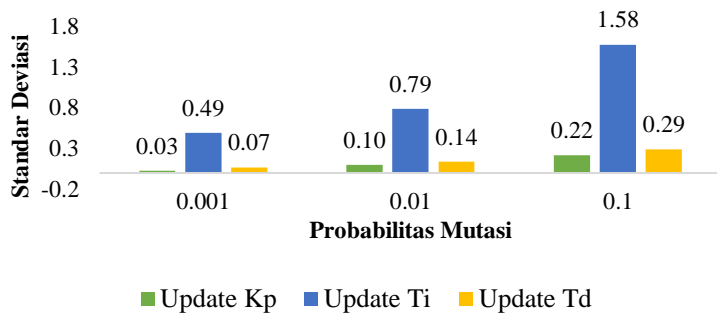
Respon level air *steam drum* ketika probabilitas mutasi dirubah, memiliki sedikit pengaruh pada spesifikasi respon sistem. Hal ini dapat dilihat dari spesifikasi respon sistem setelah dilakukan pengujian seperti pada Tabel 4.15. Berdasarkan hasil simulasi, ketika nilai probailitas mutasi yang digunakan sebesar 0,001 memiliki indeks performansi kesalahan, *perturbation peak*, dan *recovery time* yang tidak berbeda jauh ketika probabilitas mutasinya 0,01 atau 0,1. Namun, berdasarkan pengamatan pada perubahan nilai *update* parameter kontroler diperoleh kesimpulan bahwa semakin besar nilai probabilitas mutasi, maka semakin besar standar deviasi *update* parameterternya seperti data pada Gambar 4.31.



Gambar 4.30. Respon Level Air *Steam Drum* Menggunakan Kontroler PID-GA dengan Variasi Probabilitas Mutasi

Tabel 4.15. Spesifikasi Respon Sistem Menggunakan Kontroler PID-GA dengan Variasi Probabilitas Mutasi

Parameter GA	ISE	IAE	ITAE	ρ (m)	t_{rs} (detik)
PMut = 0.001	25,79	28,68	2662	0,188	285,9
PMut = 0.01	25,83	28,76	2681	0,188	2863
PMut = 0.1	27,29	29,56	2770	0,187	290,2



Gambar 4.31. *Update* Parameter Kontroler PID-GA dengan Variasi Probabilitas Mutasi

BAB 5

KESIMPULAN

Kontroler PID-GA mampu memenuhi spesifikasi level air *steam drum* yang diinginkan, yaitu memiliki *perturbation peak* kurang dari 0,25 m dan *recovery time* kurang dari 350 detik, ketika terdapat laju aliran uap keluar dari drum.

Kontroler PID-GA mampu meredam gangguan pada sistem. Semakin besar beban pada sistem, maka *perturbation peak* akan semakin besar. Ketika sistem diberi beban minimal, nominal, dan maksimal, maka diperoleh *perturbation peak* masing-masing 0,16 m; 0,19 m; dan 0,23 m.

Jumlah generasi pada kontroler PID-GA semakin besar, waktu komputasi yang dibutuhkan semakin lama. Ketika jumlah generasi yang digunakan sebanyak 10; 100; dan 1000, maka memiliki waktu komputasi masing-masing sebesar 71 detik, 600 detik, dan 5839 detik.

Probabilitas mutasi pada kontroler PID-GA semakin besar, standar deviasi dari *update* parameter kontroler PID semakin besar. Standar deviasi komponen K_p untuk probabilitas *crossover* 0,2; 0,6; dan 0,8 adalah sebesar 0,03; 0,10; dan 0,22. Komponen τ_i memiliki standar deviasi sebesar 0,49; 0,79; dan 1,58. Standar deviasi komponen τ_d yaitu 0,07; 0,14; dan 0,29.

Probabilitas *crossover* pada kontroler PID-GA tidak berpengaruh secara signifikan terhadap respon sistem. Ketika probabilitas *crossover* dipilih 0,2; 0,4; dan 0,8, diperoleh *perturbation peak* masing-masing 0,177 m; 0,183 m; dan 0,177 m dengan *recovery time*, yaitu 313 detik

Halaman ini sengaja dikosongkan

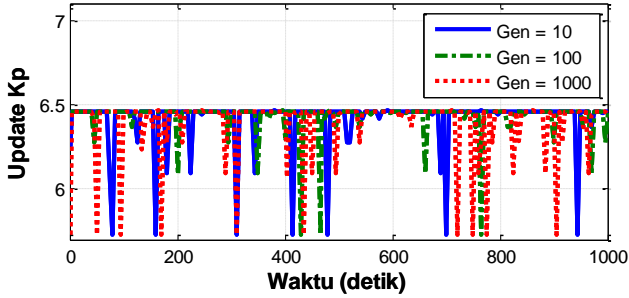
DAFTAR PUSTAKA

- [1] K. D. P. and N. I. J., "*Power System Engineering*", New Delhi: Tata McGraw-Hill, 2008.
- [2] B. A. Ogunnaike and W. H. Ray, "*Process Dynamics, Modeling, and Control*", New York: Oxford University Press, 1994.
- [3] G. Gilman, "*Boiler Control Systems Engineering*", United States of America: International Society of Automation, 2010.
- [4] A. K. J. and H. Tore, "*Advanced PID Control*", United States of America: Instrumentation, Systems, and Automation Society, 2006.
- [5] --, "*Analog Functional Diagram of Feedwater Control*", Indonesia: ABB Combustion Engineering, 1990.
- [6] A. Altinten, F. Ketevanlioglu, S. Erdogan, H. Hapoglu and M. Albaz, "Self-tuning PID Control of Jacketed Batch Polystyrene Reactor using Genetic Algorithm," *Chemical Engineering Journal*, vol. 138, pp. 490-497, 2008.
- [7] A. Jayachitra and R. Vinodha, "Genetic Algorithm Based PID Controller Tuning Approach for Continuous Stirred Tank Reactor," *Advances in Artificial Intelligence*, pp. 1-8, 2014.
- [8] T. Teng, J. Shieh and C. Chen, "Genetic Algorithms Applied in Online Autotuning PID Parameters of a Liquid-level Control System," *Transactions of the Institute of Measurement and Control*, vol. 25, no. 5, pp. 433-450, 2003.
- [9] --, "*Sistem Pengaturan Boiler PT. PJB UP 1 dan 2 Paiton*", Paiton, 2014.
- [10] --, "*Design and Operation Manual Volume 2 Paiton Steam Power Plant Units 1 and 2*", America: ABB Combustion Engineering, 1993.
- [11] R. M. Felder, R. W. Rousseau and L. G. Bullard, "*Elementary Principles of Chemical Processes*", 4th ed., America: John Wiley & Sons, 2015, p. 93.
- [12] --, "*Boilers*", [Online]. Available: <http://pudeb.com/heavy-lifting-using-hydraulic-jacks-installing-a-boiler-drum/>. [Accessed 19 December 2016].

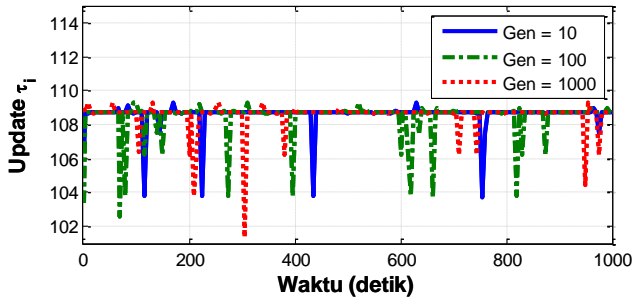
- [13] N. Jain and A. Tiwari, "Comparative Study of Fluid Coupling for Oil and Water as Working Fluid," *International Journal of Engineering Research and Development*, vol. 9, no. 6, pp. 56-65, 2013.
- [14] A. A. A. Emhemed, R. B. Mamat and D. Hanafi, "Comparison of Boiler Plant Model and Real Plant Responses Based on Proportional Integral Controller," in *IEEE Conference on Sustainable Utilization and Development in Engineering and Technology (STUDENT)*, Kuala Lumpur, 2012.
- [15] G. J. Silva, A. Datta and S. Bhattacharyya, "*PID Controllers for Time-Delay System*", Birkhauser Boston: USA, 2015.
- [16] K. Ogata, "*Modern Control Engineering*", 4th ed., Prentice Hall, 2002.
- [17] V. U. Bhaksi and U. A. Bhaksi, "*Control System Engineering*", Pune: Technical Publication Pune, 2008.
- [18] Chipperfield, "Introduction to genetic algorithms," in *Genetic Algorithms in Engineering Systems*, Stevenage, *IEE Engineering Series*, 1997, pp. 1-41.
- [19] Y. Chen and Y.-J. Y. W.-x. Ma, "Application of Improved Genetic Algorithm in PID Controller Parameters Optimization," *Telkomnika*, pp. 1524-1530, 2013.

LAMPIRAN

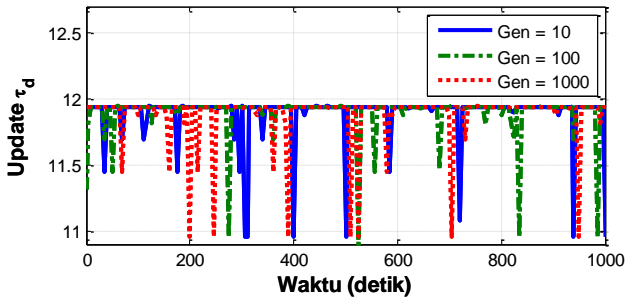
Lampiran 1. *Update* Parameter dengan Variasi Jumlah Generasi



Gambar 1. *Update* K_p dengan Variasi Jumlah Generasi

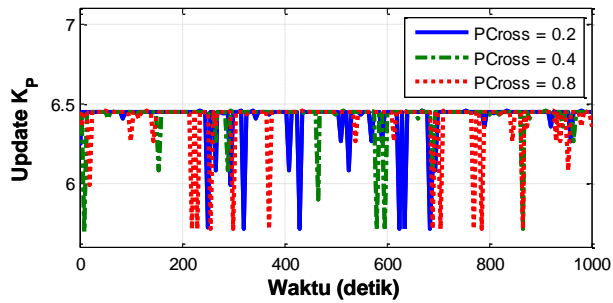


Gambar 2. *Update* τ_i dengan Variasi Jumlah Generasi

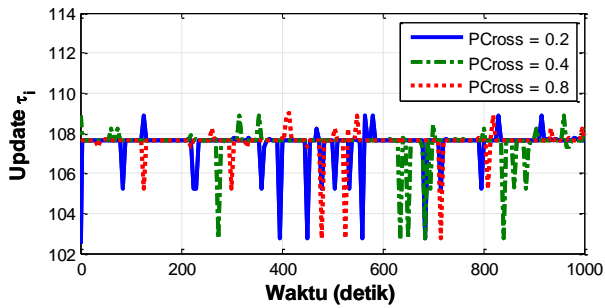


Gambar 3. *Update* τ_d dengan Variasi Jumlah Generasi

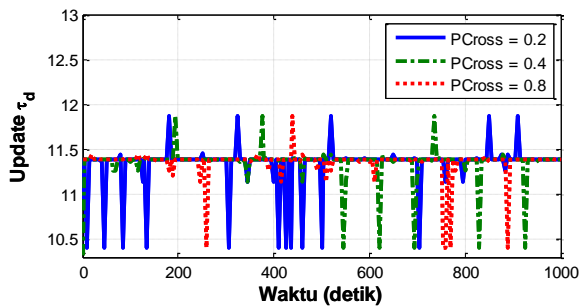
Lampiran 2. *Update* Parameter dengan Variasi Probabilitas *Crossover*



Gambar 4. *Update* K_p dengan Variasi Probabilitas *Crossover*

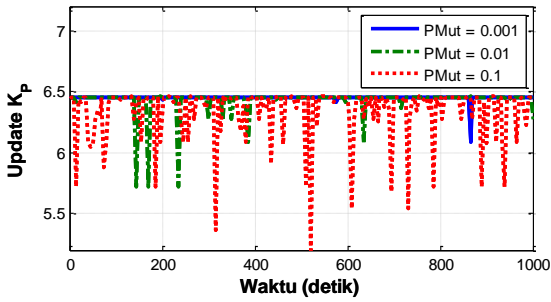


Gambar 5. *Update* τ_i dengan Variasi Probabilitas *Crossover*

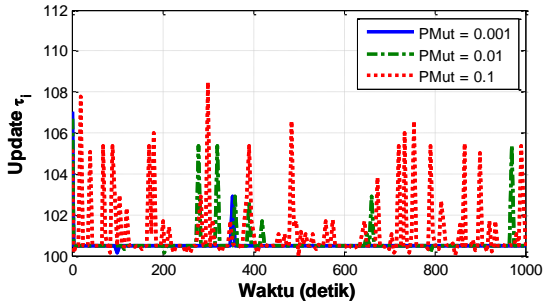


Gambar 6. *Update* τ_d dengan Variasi Probabilitas *Crossover*

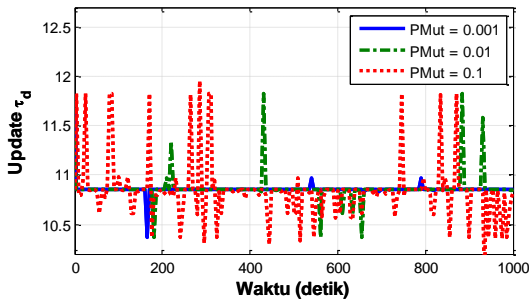
Lampiran 3. *Update* Parameter dengan Variasi Probabilitas Mutasi



Gambar 7. *Update* K_p dengan Variasi Probabilitas *Crossover*



Gambar 8. *Update* τ_i dengan Variasi Probabilitas *Crossover*



Gambar 9. *Update* τ_d dengan Variasi Probabilitas *Crossover*

Lampiran 4. Program Kontroler PID-GA

```
function out = GA1(x)
global ek t JJJJ Kp1 Ti1 Td1 Vr1 minKp1 maxKp1 minTi1 maxTi1
minTd1 maxTd1 krom1 ukpop1 ...
    uksol1 ukbit1 ukkro1 gen1 maxgen1 pop1 popbin1 popdec1 fitness1
turselect1 ...
    ortuselect1 valortu1 turpop1 probcross1 sortin1 psort1 couponum1
perukkro1 ...
    coupcross1 crosselector1 probmut1 popmut1 mutsum1 mutbar1
mutkoll1 popoptim1 ...
    popoptimbin1 popoptimdec1 ...
%=====
% Insialisasi parameter "GA & PID"
%=====
ek          = x(1);
JJJJ        = x(2);
t           = x(3);

if t==0
    ukpop1      = 30;
    ukkro1      = 30;
    pop1        = rand(ukpop1,ukkro1)<0.5;
    popbin1     = double(pop1)';
    minKp1      = 1;
    maxKp1      = 10;
    minTi1      = 60;
    maxTi1      = 1000;
    minTd1      = 1;
    maxTd1      = 13;
    out(1)      = minKp1 + (maxKp1-minKp1).*rand(1);
    out(2)      = minTi1 + (maxTi1-minTi1).*rand(1);
    out(3)      = minTd1 + (maxTd1-minTd1).*rand(1);
end

ukpop1      = 30;
ukkro1      = 30;
uksol1      = 3;
ukbit1      = ukkro1/uksol1;
```

```

gen1      = 0;
maxgen1   = 40;
probcross1 = 0.6;
probmut1  = 0.02;
popdec1   = [];
Vr1       = 100;
minKp1    = 5;
maxKp1     = 20;%80
minTi1    = 100;
maxTi1     = 200;%6000
minTd1    = 10;
maxTd1     = 30;%150

```

```

while gen1<maxgen1
    gen1      = gen1+1;
    for i = 1:ukpop1
        for j = 1:uksol1
            krom1      = 1:ukbit1:ukkro1;
            popdec1(j,1) = polyval(popbin1(i,krom1(j):ukbit1*j),2);
        end
        Kp1      = popdec1(1)/(2^ukbit1)*Vr1;
        Ti1      = popdec1(2)/(2^ukbit1)*Vr1;
        Td1      = popdec1(3)/(2^ukbit1)*Vr1;
        out(1)    = minKp1+((maxKp1-minKp1)/((2^ukbit1)-1))*Kp1;
        out(2)    = minTi1+((maxTi1-minTi1)/((2^ukbit1)-1))*Ti1;
        out(3)    = minTd1+((maxTd1-minTd1)/((2^ukbit1)-1))*Td1;
        fitness1(i,1) = 1/(1+JJJJ);
    end
    %=====
    % Persyaratan Kromosom Terbaik
    %=====
    if (gen1 == maxgen1)|(ek=10^(-3))
        break
    end
    %=====
    % Operator Seleksi : Tournament
    %=====
    for i=1:ukpop1
        for j=1:2

```

```

        turselect1 = randi(ukpop1,1);
        ortuselect1(j,:) = pop1(turselect1,:);
        valortu1(j,1) = polyval(ortuselect1(j,:),2);
    end
    if valortu1(1)>=valortu1(2)
        turpop1(i,:) = ortuselect1(1,:);
    else
        turpop1(i,:) = ortuselect1(2,:);
    end
end
pop1 = turpop1;
%=====
% Operator Crossover : Single Point
%=====
[~,sortin1] = sort(rand(size(pop1 ,1) ,1));
psort1 = pop1(sortin1, :);
coupnum1 = size(psort1, 1)/2;
perukkro1 = size(psort1, 2);
coupcross1 = randi(coupnum1, 1) < probccross1;
croselector1 = coupcross1.*randi([1,perukkro1],[coupnum1,
1]);
for i = 1:coupnum1
    pop1([2*i-1,2*i],:) = [psort1([2*i-1,2*i],1:croselector1(i)),...
        psort1([2*i,2*I 1],croselector1(i)+1:perukkro1)];
end
%=====
% Operator Mutasi : Bit Flip
%=====
popmut1 = pop1;
mutsum1 = ceil((ukpop1-1)*ukkro1*probmut1);
mutbar1 = ceil(rand(1,mutsum1)*(ukpop1-1))+1;
mutkol1 = ceil(rand(1,mutsum1)*ukkro1);
for i = 1:mutsum1
    popmut1(mutbar1(i),mutkol1(i)) =
abs(popmut1(mutbar1(i),mutkol1(i))-1);
end
[fitness1,ind1] = max(fitness1);
popoptim1 = popmut1(ind1,:);
popoptimbin1 = double(popmut1)';

```

```

for j = 1:uksol1
    krom1 = 1:ukbit1:ukkro1;
    popoptimdec1(j,1) = polyval(popoptimbin1(j,krom1(j):ukbit1*j),2);
end
popbin1 = popmut1;
Kp1 = popoptimdec1(1)/(2^ukbit1)*Vr1;
Ti1 = popoptimdec1(2)/(2^ukbit1)*Vr1;
Td1 = popoptimdec1(3)/(2^ukbit1)*Vr1;
out(1) = minKp1+((maxKp1-minKp1)/((2^ukbit1)-1))*Kp1;
out(2) = minTi1+((maxTi1-minTi1)/((2^ukbit1)-1))*Ti1;
out(3) = minTd1+((maxTd1-minTd1)/((2^ukbit1)-1))*Td1;
end
end

```

Lampiran 5. Program Indeks Performansi Kesalahan

```

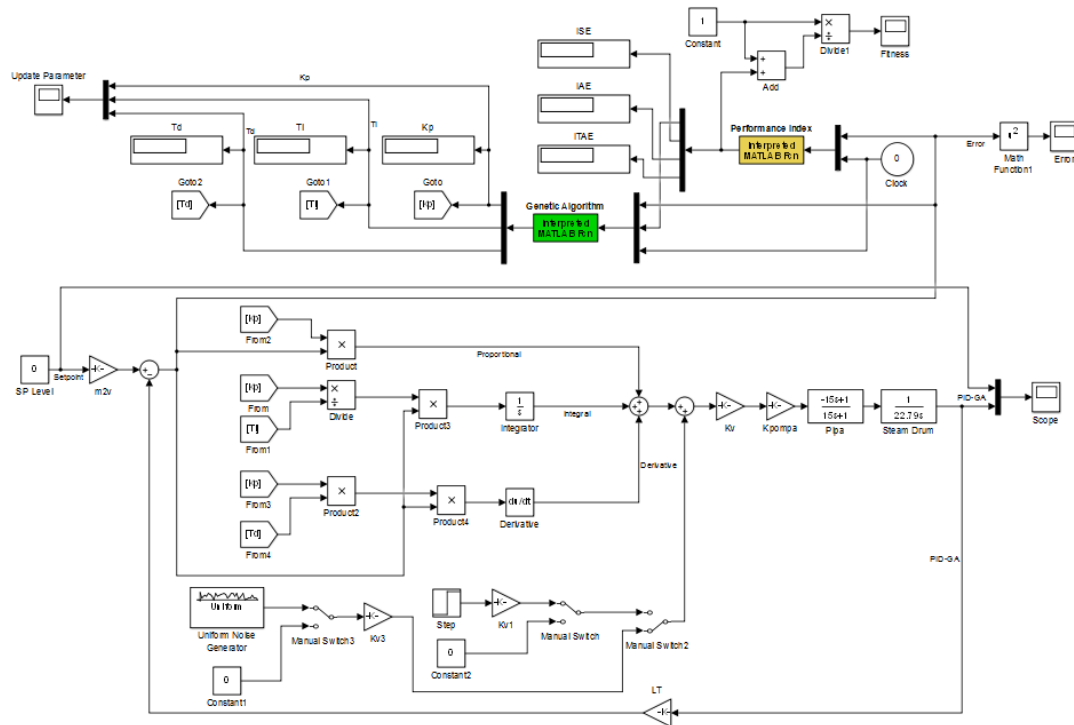
function out = PIGA(x)
global J1 J2 J3 w1 w2 w3 J ek t
ek = x(1);
t = x(2);
if t == 0
    J1 = 0;
    J2 = 0;
    J3 = 0;
end

w1 = 0.5;%0.5
w2 = 0.2;%0.2
w3 = 0.3;%0.3
J1 = J1+((ek)^2); %ISE
J2 = J2+(abs(ek)); %IAE
J3 = J3+(t*abs(ek)); %ITAE
J = (w1*J1)+(w2*J2)+(w3*J3); %Ind Performa
out(1) = J;
out(2) = J1;
out(3) = J2;
out(4) = J3;

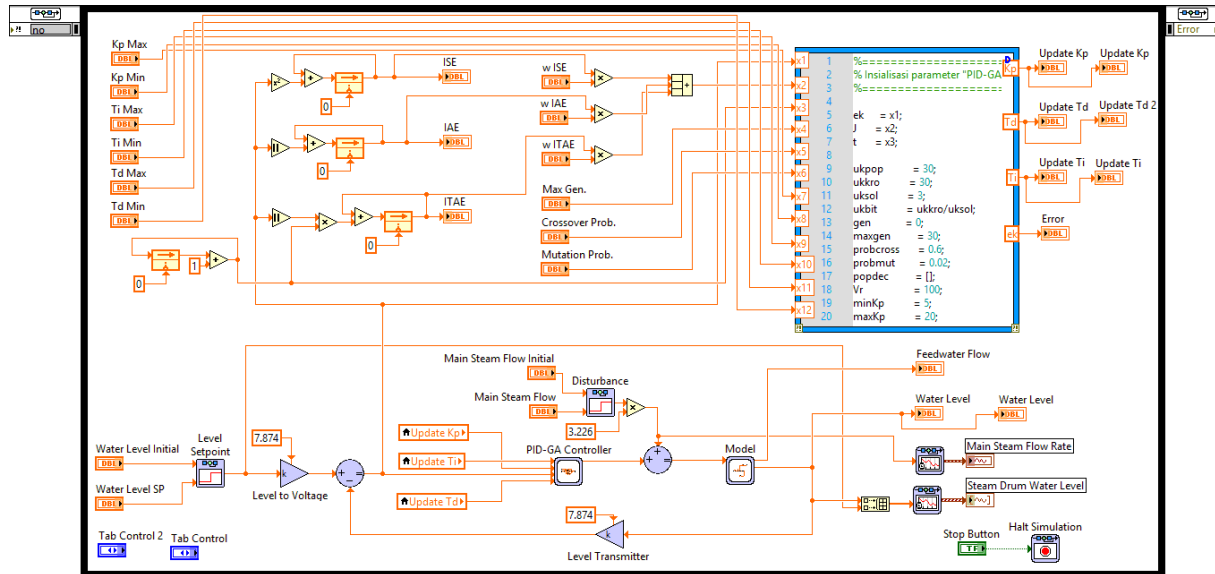
end

```

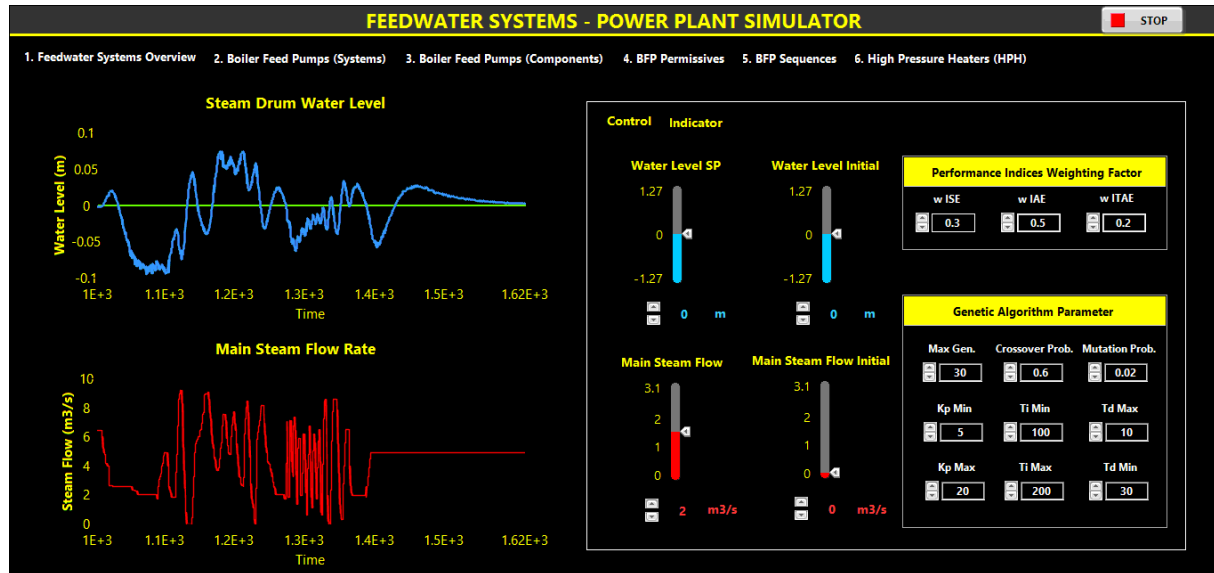
Lampiran 6. Simulink Kontroler PID-GA



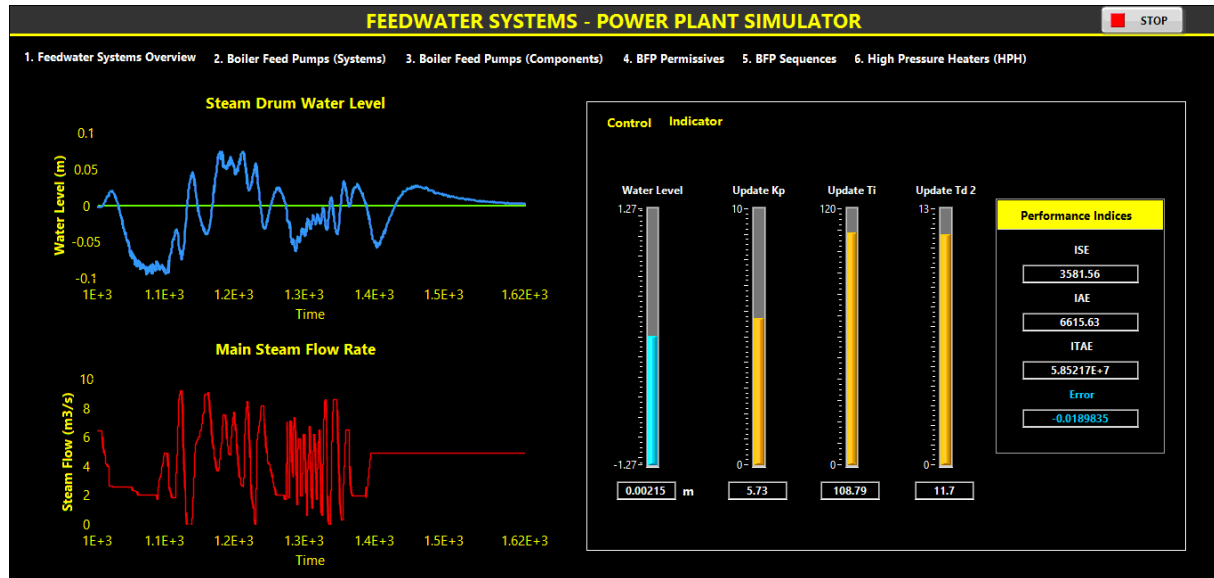
Lampiran 4. Program *Virtual Plant: Feedwater Systems*



Lampiran 4. Tampilan Menu *Control Virtual Plant: Feedwater Systems*



Lampiran 4. Tampilan Menu *Indicator Virtual Plant: Feedwater Systems*



Halaman ini sengaja dikosongkan

RIWAYAT PENULIS



Mohamad Yusuf lahir di kabupaten Probolinggo, provinsi Jawa Timur pada tanggal 25 April 1995 dari pasangan Mohamad Nasir dan Anicha Su'ud. Penulis merupakan anak kedua dari dua bersaudara. Setelah menyelesaikan Sekolah Dasar, Sekolah Menengah Pertama dan Sekolah Menengah Atas di daerah asalnya yaitu Probolinggo pada tahun 2013, dengan izin Allah SWT serta kedua orang tua, penulis melanjutkan pendidikan di Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya. Penulis mempunyai semangat dan kemauannya yang keras untuk selalu menjadi lebih baik dan mempunyai prinsip hidup yaitu "Tidak ada kata tidak bisa selagi kita mau berusaha dan berdoa kepada Tuhan Yang Maha Esa". Kata-kata tersebut didapatkan dari Almarhummah Ibu tercintanya dalam memberi motivasi dikala penulis merasa kesusahan.

E-mail: 25.mohamadyusuf@gmail.com